Universidad Carlos III de Madrid
Escuela Politécnica Superior

Departamento de Teoría de la Señal y Comunicaciones
Doctorado en Multimedia y Comunicaciones

Tesis Doctoral

Análisis Exploratorio de Datos de Expresión Genómica mediante el Análisis en Conceptos Formales

José María González Calabozo

Dirigida por

Francisco José Valverde Albacete

Carmen Peláez Moreno

Leganés, 2016

E-mail: jmgcalabozo@gmail.com

Address:

Grupo de Procesado Multimedia
Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid
Av. de la Universidad, 30
Leganés 28911 — Spain

Análisis Exploratorio de Datos de Expresión Genómica mediante el Análisis en Conceptos Formales

Autor: José María González Calabozo
Directores: Francisco José Valverde Albacete
Carmen Peláez Moreno

Firma del Tribunal Calificador:

	Nombre y Apellidos	Firma
Presidente: D.	.....	.....
Vocal: D.	.....	.....
Secretario: D.	.....	.....

Calificación: .....

Leganés, ... de ..... de 2016.

Índice general

Índice de figuras

Índice de tablas

Abstract

Gene Expression Data (GED) analysis poses a great challenge to the scientific community that can be framed into the Knowledge Discovery in Databases (KDD) and Data Mining (DM) disciplines.

In this thesis we put forward a framework in which GED analysis is understood as an Exploratory Data Analysis (EDA) process where, by means of the adoption of Formal Concept Analysis (FCA)-related techniques, we provide support for human interaction with data aiming at improving the step of hypothesis abduction. In this way, the contributions of this thesis focuses on the adaptation to human cognition of data interpretation and visualization and the results of a DM process.

In particular, we have applied these strategies to transcriptomics, where co-clustering is usually the technique of choice. In this thesis, we do not merely introduce a co-clustering algorithm but instead we offer a set of analysis tools that revolve around $\mathcal{K}$-Formal Concept Analysis, a generalization of FCA that allows to consider real-valued matrices. By using either max-plus or min-plus as the underlying semirings, we obtain interpretations for gene *under- and over-expression* respectively, thereby also introducing the notion of *threshold of expression*, a value that determines how the GED matrix is transformed into a concept lattice.

In this way, the GED analysis problem gets transformed into the exploration of a sequence of lattices indexed by the aforementioned threshold, enabling the visualization of the hierarchical structure of the co-clusters with a certain degree of granularity. Our graphical representation of this sequence ensures that all the co-clusters with the same set of conditions are always plotted in the same spatial coordinates, therefore facilitating their interpretation and allowing us to introduce the notion of *persistence* or *robustness* of a co-cluster.

On the other hand, the resulting conceptual lattice can be used to index external databases, such as Gene Ontology (GO), thus offering a new way of accessing other available resources. In this setting, the sequence of lattices from a particular experiment indexes or vertebrates the researcher vision of that given resource. This also allows us:

- To obtain a quality measure of the co-clusters by obtaining their p-values according to the terminology of those resources,
- To observe the evolution of a gene throughout the different formal concepts it appears in, as the threshold of expression is modified, including ample information about its characteristics as provided by those resources, and
- To look for formal concepts or relevant co-clusters observing which genes are included and what their persistence is, to infer, for example, hypotheses on their function.

We illustrate the exploration procedure with two real data examples: the effects of selenium on Arabidopsis Thaliana and the response of human trisomic cells to doxycycline.

Resumen

El análisis de Datos de Expresión Genética (ing. "Gene Expression Data", GED) supone un gran reto para la comunidad científica que, debido a sus características, podemos enmarcar en las disciplinas de Descubrimiento de Conocimiento en Bases de Datos (ing. "Knowledge Discovery in Databases", KDD) y Minería de Datos (ing. "Data Mining", DM).

En esta tesis proponemos un sistema en el que entendemos el análisis de GED como un proceso Análisis Exploratorio de Datos (ing. "Exploratory Data Analysis", EDA) y en el que mediante la adopción de técnicas basadas en el Análisis en Conceptos Formales (ing. "Formal Concept Analysis", FCA) proporcionamos soporte para la interacción humana con los datos, con el objetivo de mejorar el proceso de abducción de hipótesis. Así, las contribuciones de esta tesis se centran en la adaptación a la cognición humana de la interpretación y visualización de los datos y resultados del proceso de DM.

En concreto, el dominio de conocimiento en el que se han aplicado estas estrategias es el de la transcriptómica en el que la *co-agrupación* (o *co-clustering*) de genes es el enfoque más comúnmente adoptado. En esta tesis no planteamos simplemente un algoritmo de co-agrupamiento sino un conjunto de herramientas de análisis que giran en torno a $\mathcal{K}$-Formal Concept Analysis una generalización de FCA que permite estudiar matrices en el dominio de los números reales. Utilizando como semi-anillos subyacentes las álgebras maxplus y minplus se obtienen interpretaciones de la *infra-expresión* y la *sobre-expresión* de los genes, respectivamente introduciendo además la noción de umbral de expresión, un valor que determinará cómo se transforma la matriz de expresión genética (GED) en un retículo de conceptos.

De esta manera, el problema del análisis de GED se transforma en la exploración de una *secuencia de retículos* indexados por dicho umbral que permiten visualizar la estructura jerárquica de los co-agrupamientos con mayor o menor nivel de granularidad. Nuestra representación gráfica de esta secuencia permite comparar cómo varían los retículos de conceptos dibujando siempre los conceptos que involucran al mismo conjunto de condiciones en la misma posición, lo que facilita su interpretación e introduce el concepto de *persistencia* o *robustez* de un co-agrupamiento.

Por otra parte, el retículo conceptual resultante del FCA puede usarse para indexar bases de datos externas lo que ofrece una nueva manera para acceder a otros recursos disponibles como Gene Ontology (GO), en dónde la secuencia de retículos resultante de un experimento particular indexa o vertebra la visión del investigador de dicho recurso. Además esto nos permite:

- obtener una medida de la calidad de los co-agrupamientos mediante el p-valor obtenido a la hora de analizar las terminología de estos recursos,

- observar la evolución de un gen a través de los diferentes conceptos formales en los que aparece a medida que se modifica el umbral, contando con amplia información acerca de las características del gen proporcionada por estos recursos, y
- buscar conceptos formales o co-agrupamientos de interés y ver qué genes están incluidos en función del umbral de confianza aplicado para inferir, por ejemplo, hipótesis sobre su función.

Ilustramos este procedimiento con el análisis de datos reales de los efectos del selenio en la Arabidopsis Thaliana y de la respuesta a la doxiciclina de células trisómicas humanas.

Agradecimientos

En primer lugar me gustaría agradecer a mis padres por toda la ayuda que me han brindado, por el interés que han mostrado en que siempre siga aprendiendo. Me han proporcionado el apoyo que he necesitado, en especial para acabar esta tesis.

A Fran y Carmen mis directores de tesis: claramente han sido de gran ayuda. Han comprendido lo difícil que ha sido para mi compaginar un trabajo a tiempo completo con un doctorado, sobretodo en este último año conmigo trabajando fuera de España.

No puedo dejar sin mencionar a Bea, Juanjo, Julio y Rober por esas tardes que nos reuníamos en Madrid y hablábamos de todo un poco. Muy buenas ideas surgían de esas conversaciones, algunas de las cuales sirvieron de inspiración para alguna parte de esta tesis.

También agradecer a esas personas con las que he compartido más tiempo durante este último año, mis compañeros de trabajo en Munich, Elena, Eva, Tamara, Beatriz, Nando, Mariano, Jesús, Juli, Manuel, Patri, Cristina, Eva, Angel, Nacho y Antonio. Juntos hemos pasados buenos y no tan buenos momentos en este nuevo proyecto. Me quedo con los viajes por distintos pueblos de Baviera y ese rico Schnitzel o codillo acompañado de una Weißbier.

Por último a mis amigos de Móstoles, Almu, Angi, Ana, Angel, Carlos, Carlos, Carlos, Cris, Fonta, Inés, Irene, Jaime, Javi, Jimmy, Juanjo, Laura, Mayte, Miriam, Oscar, Raquel, Ricky, Sara, Santi, Sergio, Vero, a todos ellos que ya no admitirán como excusa válida el estar trabajando en el doctorado para no salir a tomar algo cualquier día.

José María González Calabozo
Leganés, febrero de 2016

1

Introducción

Estamos viviendo en la era de la información donde está a nuestra disposición una gran cantidad de datos, muchos más de los que podemos procesar de forma manual. Por este motivo existen cientos de técnicas de procesamiento de datos que ayudan a los investigadores a encontrar el conocimiento que necesitan entre toda la información disponible.

En particular, en los últimos años se viene produciendo una revolución en el campo de la genética: nuevas técnicas de secuenciación, medicamentos específicos para cada persona, tratamientos contra el cáncer, etc... La ingeniería genética abre un mundo de posibilidades a la hora de tratar estos temas, pero para ello primero es necesario conocer exactamente cuál es la función de los genes de un organismo y saber cómo los genes interactúan entre sí.

Una forma de hacerlo es viendo la respuesta de los genes de un organismo en diferentes situaciones. Es aquí donde entra en juego la epigenética, ciencia que se encarga del estudio de la actividad de los genes. Incluso la célula más simple posee miles de genes que se utilizan para fabricar proteínas mediante la transcripción hecha por el ARNm en lo que se conoce como **dogma central de la biología molecular**[1] [2].

Con métodos desarrollados en las últimas tres décadas es posible muestrear las concentraciones de los productos de expresión de miles de genes a la vez mediante el muestreo de las concentraciones de ARNm y obtener así datos de expresión genética (en sus siglas en inglés, "Genetic expression data" GED), aunque los datos así obtenidos son masivos y densos.

En esta tesis se propone un conjunto de herramientas de análisis de datos que, centrándose en el Análisis en Conceptos Formales, lo amplia de diversas formas, dándole al investigador la posibilidad de explorar los datos y tratar de darles significado, por ejemplo.

El dominio de conocimiento en el que hemos decidido aplicar este análisis de datos es el de la transcriptómica nivel de expresión de Ácido ribonucleico (ARN) mensajero en cada una de las diferentes muestras. Mediante un sistema de exploración se podrán encontrar los genes que parecen tener un comportamiento similar y gracias a la conexión con bases de datos externas se podrá contrastar o ampliar el conocimiento que surge de dicha agrupación.

Estas herramientas se pueden englobar dentro del análisis exploratorio de datos donde no se realiza ninguna suposición acerca de la distribución de los datos de entrada y el propio análisis se enfoca en la dirección adecuada a medida que se va procesando más la información. El

[1] Para una explicación más detallada del *teorema fundamental de la biología molecular* y las diferentes técnicas para medir el nivel de concentración de cadenas de ARNm se recomienda consultar el Apéndice A.

objetivo principal del análisis se centra en el co-agrupamiento, de los elementos representados mediante una matriz teniendo en cuenta tanto filas como columnas. Con este fin ha puesto especial hincapié en diferentes modos de visualización de los co-agrupamientos de forma que, al ver los datos representados de diferentes puntos de vista, se consiga extraer el conocimiento deseado. Además se alienta al uso de bases de datos externas que ayudan a mejorar el resultado del co-agrupamiento.

En este primer capítulo se hará una introducción al estado actual del análisis exploratorio de datos y cómo la metáfora de los *paisajes del conocimiento* se puede aplicar al análisis en conceptos formales, en particular, en el contexto del análisis de los datos de expresión genética. En el segundo capítulo se hace un resumen del estado del arte de los diferentes algoritmos de agrupamiento que se han estudiado en el campo de la transcriptómica. El tercer capítulo se centra en explicar las técnicas desarrolladas en esta tesis para, en los capítulos cuarto y quinto, mostrar varios ejemplos "in silico" y reales de cómo realizar estos análisis. En el sexto y último capítulo se encuentran las contribuciones de esta tesis, así como las conclusiones y trabajos futuros. Además se han incluido tres apéndices: uno donde se da un breve repaso al campo de la genética orientado a la transcriptómica, otro que proporciona el trasfondo de álgebra necesario para entender algunas fórmulas usadas en esta memoria, y un último apéndice, que es una breve referencia al software desarrollado en esta tesis, una web disponible en `http://webgenekfca.com`.

A continuación, sin embargo, desarrollamos los temas introducidos en esta breve reseña.

1.1 Datos de expresión genética

Cuando se dice que "un gen se activa" significa que en la célula se crean cadenas de ARNm que portan una copia de dicho gen, cadenas que se utilizan para sintetizar una proteína en los ribosomas. Pero no todos los genes están activos a la vez: todo depende de las condiciones internas y externas de la célula. Así, por ejemplo, habrá diferentes genes activos en una neurona que en un leucocito, aunque ambas células pertenezcan al mismo organismo y compartan el mismo Ácido desoxirribonucleico (ADN). También es posible que se activen o desactiven genes como respuesta a diferentes cambios en el entorno de la célula, como por ejemplo reacciones a alguna sustancia tóxica como se muestra en el ejemplo de la Sección 5.2.

El nivel de expresión de un gen en un momento dado viene dado por la cantidad sus moléculas de ARNm que se encuentran en la célula en ese momento. Analizando el nivel de expresión de miles de genes en células similares pero en muestras sometidas a diferentes condiciones se puede ver cómo éstas afectan a las células.

Existen distintas formas de medir el nivel de expresión genética de un conjunto de células. Durante lustros la opción más popular sin duda ha sido la técnica de microarrays [3], pero las técnicas basadas en RNA sequencing (RNA-Seq) están abaratando sus costes rápidamente [4].

Comoquiera que se obtenga el nivel de expresión genética, al final, para cada muestra, se generará un vector con el valor de expresión genética para cada gen. Este valor puede ser una concentración relativa respecto a otro nivel o el conteo de cadenas de ARNm: en esta fase del proceso esta consideración no es importante.

Y este proceso se repite para diferentes muestras en diferentes condiciones en función de los objetivos de estudio. Cuando se tienen los niveles de expresión de miles de genes de las diferentes muestras se genera una matriz llamada GED $M \in \mathcal{R}^{g \times m}$ donde la fila $i \in 1...g$ representa el nivel de expresión del gen i en la muestra $j \in 1...m$.

Es a partir de esta matriz GED cuando empieza el agrupamiento de genes y muestras en función de su nivel de expresión.

1.2 Análisis Exploratorio de Datos

El análisis exploratorio de datos (por sus siglas en inglés, Exploratory Data Analysis (EDA)) es una filosofía de análisis de datos que se contrapone al análisis confirmatorio (ing. Confirmatory Data Analysis (CDA)), y que preconiza el uso de estadísticos de agregación y diagramas de visualización de los datos previos al enunciado de hipótesis en el método estadístico como una forma de guiar la concepción de hipótesis antes de su comprobación. Tukey fue uno de los primeros que propuso utilizar esta forma de análisis, que dejó bastante abierta, no llegando a dar una definición exacta [5], aunque sí bastantes ejemplos y herramientas[2].

El análisis confirmatorio proporciona información precisa cuando las conjeturas realizadas se cumplen, es decir cuando el modelo de probabilidad es el adecuado y se analizan las hipótesis correctas. Por contra el EDA propone un conjunto de técnicas que permiten obtener una visión más general de la estructura de los datos, para posteriormente poder centrarse en los detalles deseados. Preconiza, además, una representación gráfica fácil de entender, como podría ser por ejemplo un diagrama de cajas (ing. "box plot"). A diferencia del clásico análisis confirmatorio, en EDA no se hacen suposiciones acerca de los datos, con lo cual se consigue una mayor flexibilidad.

Se ha llegado a decir que EDA no es un conjunto de técnicas sino una actitud: flexibilidad y confianza en el modelo visual [1]. Consecuentemente, antes de comenzar el *análisis de inferencia* (el procedimiento típico del análisis confirmatorio) es necesario realizar un análisis exploratorio de datos que permita descubrir las preguntas correctas, muy relacionadas con cómo se realizará el diseño.

Figura 1.1: Paradigma del funcionamiento de la ciencia e ingeniería según [1]

Por ejemplo, se puede tener la sospecha de que ciertos genes estén relacionados con el desarrollo de un tipo de cáncer y se quiere averiguar si realmente es así. Esto sería la idea correspondiente a la figura 1.1. Pero hace falta una pregunta específica que proporcione un diseño experimental específico. La pregunta que se desea responder podría ser si el cáncer modifica el nivel de expresión de ciertos genes, pero es una pregunta que no se puede responder fácilmente debido a las limitaciones del experimento: las células son muy complejas y el nivel

[2] Como el resumen de una distribución mediante cinco cifras: sus cuartiles y el rango

de expresión puede ser modificado por el cáncer o por una multitud de agentes externos o internos. Además cada individuo tiene un código genético diferente con unas manifestaciones epigenéticas diferentes en función del ambiente en el que se ha desarrollado. Por otro lado, sería prácticamente imposible la dirección de una hipotética relación causa-efecto: es decir, si la expresión del gen es la que favoreció el desarrollo del cáncer, o más bien es el cáncer quien modifica la expresión genética. Esto se podría saber sólo si se pudiera tener una observación justo antes de que se desarrollara el cáncer y se pudiera comprobar si el nivel de expresión de los genes en cuestión precede al desarrollo del cáncer o no. Así pues la pregunta que se podría plantear es *¿cuáles son los genes que presentan un nivel de expresión diferente al normal en función de si el tejido es cancerígeno o no?*. Para ello el experimento debiera contar con muestras de tejidos sanos y con cáncer de diferentes individuos. Una vez se obtiene la colección de datos se procede al EDA, donde una exploración permite ver la distribución de los datos y qué información se puede obtener de ellos. Una vez desarrollado el modelo se podría aplicar un análisis confirmatorio de datos que diera una respuesta a la pregunta formulada.

El párrafo anterior describe una situación típica en la que queremos avanzar el conocimiento científico sobre un fenómeno que acontece y que queda capturado en unos datos de medición. Esta tesis contribuye al desarrollo del análisis exploratorio de un tipo particular de datos tradicionalmente considerados: las matrices de contingencia de dos modos [6]. Para presentar dicho análisis exploratorio a continuación introduciremos un marco para el análisis de datos en forma de matrices de dos modos.

1.3 Análisis en Conceptos Formales

El Análisis en Conceptos Formales, (por sus siglas en inglés, FCA, "Formal Concept Analysis") [7] es un método de análisis de datos creado por Rudolf Wille en 1982. Durante los primeros 10 años su uso pasó desapercibido y se limitó al que le dieron sus estudiantes en Alemania. Tal origen alemán el marcará gran parte de su nomenclatura. A partir de los 90 aparece un resurgir de este método de análisis en varias disciplinas como la lingüística, psicología, inteligencia artificial o la recuperación de información [8].

Sus fundamentos son muy intuitivos: cuando tenemos elementos comparables entre sí puede aparecer una relación de orden entre ellos, que da lugar a un conjunto parcialmente ordenado (cpo). Este cpo a su vez puede ser un *retículo* si cada par de elementos posee un máximo y un mínimo (ver Apéndice B.2). Si para cada subconjunto de elementos del conjunto parcialmente ordenados existen dicho máximo y mínimo, entonces el retículo se dice *completo*.

1.3.1 Definición

Supongamos que tenemos un **conjunto** G **de objetos**[3] y otro **conjunto** M **de atributos**[4]. Supongamos, también, que entre los dos conjuntos hay definida una relación **de incidencia** booleana $I \in 2^{G \times M}$ que asocie a cada objeto g perteneciente a G una serie de atributos

[3] Se utiliza G como abreviatura de *Gegenstand*, "objeto", en alemán.
[4] Del alemán *Merkmale* que se traduce como "atributo".

de M, y viceversa. Un **contexto formal** está constituido por la tripla (G, M, I) donde G es el conjunto de objetos, M es el conjunto de atributos e I es la relación de incidencia que relaciona ambos conjuntos $I \subseteq G \times M$.

Como a menudo nos referiremos a conjuntos finitos y relaciones finitas, podemos representar la incidencia como una *matriz* con entradas booleanas con sus filas indexadas por objetos y sus columnas por atributos. Similarmente, en la *representación tabular* de la incidencia booleana como la mostrada en la Tabla 1.1 cada fila corresponde a un objeto, en este caso un animal, y cada columna es un atributo que puede tener un animal. La celda con una X significa que el animal g tiene asociado el atributo (incidente) m, por ejemplo el avestruz es un animal ovíparo.

	mamífero	ave	reptil	bípedo	plumas	placenta	ovíparo
canguro	X			X			
perro	X					X	
avestruz		X		X	X		X
velocirraptor[5]			X	X	X		X
cocodrilo			X				X
hombre	X			X		X	
ornitorrinco	X						X

Tabla 1.1: Ejemplo de relación de incidencia entre atributos y objetos para su análisis FCA.

De un conjunto de objetos se puede sacar sus atributos comunes y de un conjunto de atributos se puede saber qué objetos presentan tales atributos en común. A un conjunto de objetos junto con sus correspondientes atributos se le denomina **concepto formal**, de donde recibe la técnica su nombre. Un concepto formal del ejemplo anterior estaría formado por un subconjunto de animales junto con el subconjunto de atributos asociados, por ejemplo: ({hombre,canguro}, {mamífero, bípedo}). La palabra "contexto" que usa la técnica se refiere, pues, al hecho de que la definición que queda hecha de "hombre" o "canguro" sólo tiene en cuenta los atributos considerados en este *contexto*, en este caso "mamífero", "bípedo", etc.

Al conjunto de objetos que pertenecen a un concepto formal se le llama **extensión** y al conjunto de atributos **intensión**[6]. Se puede establecer una relación de orden parcial entre los conceptos en función de su generalidad: un concepto con mayor número de atributos, es más **específico**, es decir estará por debajo de un concepto más **genérico**. Así pues el concepto ({hombre}, {mamífero, bípedo, placentario}) será inferior al concepto ({hombre,canguro}, {mamífero, bípedo}).

Se puede demostrar [9] que el conjunto de los conceptos formales, ordenados mediante esta relación es un retículo completo, denominado **retículo conceptual**, que se puede representar con un *diagrama de Hasse* o *de orden*(ver Apéndice B.1.2) como se muestra en la Figura 1.2, en este caso en la representación usada por el software ConExp [10]. En este diagrama se ve cómo los conceptos son más específicos si se navega hacia abajo y más generales cuando se navega hacia arriba. Más información acerca de retículos y conjuntos parcialmente ordenados se puede consultar en el Apéndice B.2.

[5] Para el ejemplo consideraremos al velocirraptor como un reptil. En 2007 se descubrió que el velocirraptor

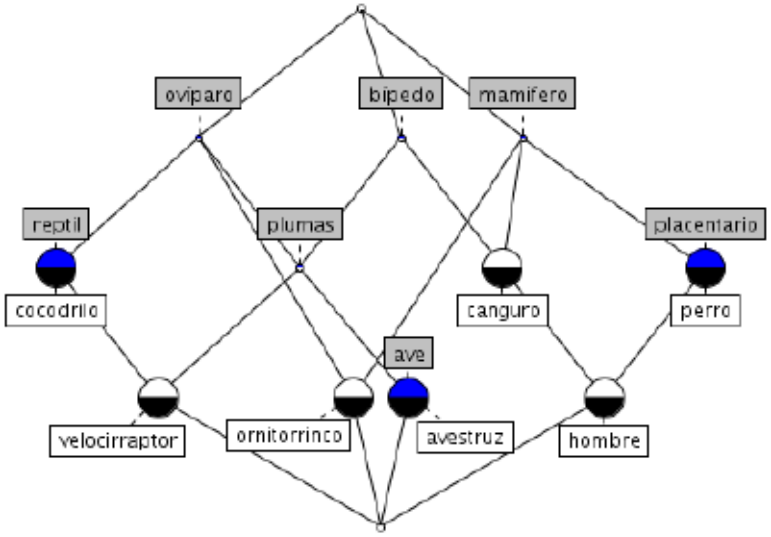


Figura 1.2: Retículo FCA de la tabla 1.1, en la representación de ConExp.

1.3.2 Propiedades

Para $A \subseteq G$ y $B \subseteq M$ los *(mapas) o funciones polares* se definen como [11, p.66]:

$$A' = \{m \in M | (g, m) \in I \ \forall g \in A\} \tag{1.3.1a}$$
$$B' = \{g \in G | (g, m) \in I \ \forall m \in B\} \tag{1.3.1b}$$

Esto significa que A' es el conjunto de atributos comunes a todos los objetos de A y B' es el conjunto de objetos que poseen los atributos de B.

Nótese que dados un par de subconjuntos cualesquiera (A, B) de objetos A y atributos B no está garantizado que formen un concepto formal: en general, cada uno de estos subconjuntos genera un concepto formal diferente, (A'', A') y (B', B''). Denotamos al **conjunto de conceptos formales**[7] por:

$$\mathfrak{B}(G, M, I) = \{(A, B) \in G \times M \mid A' = B \wedge B = A'\} \tag{1.3.2}$$

Estas dos funciones de conjuntos aplicadas sobre los subconjuntos de G y M establecen una **conexión de Galois**, particularizada por la relación dada $I \subseteq G \times M$ tal y como se explica en la sección B.3.2. Las conexiones de Galois establecen relaciones de orden paralelas (isomorfas) y contrarias en dos dominios aparentemente diferentes (en este caso los conjuntos de objetos y atributos, en principio no ligados) y aclaran de forma considerable las propiedades de las

tenía plumas.

[6] Inspirándose en la nomenclatura propuesta por Frege para conceptos similares en la Semántica denotativa.

[7] La inicial en gótico "B" viene del alemán "Begriff": "concepto".

extensiones e intensiones *intra* e *inter* dominios de definición. En particular, las siguientes propiedades de los conceptos formales (A, B) son inmediatas:

(i) $A \subseteq A'' \longleftrightarrow B \subseteq B''$

(ii) $A' = A''' \longleftrightarrow B' = B'''$

(iii) $A \subseteq B' \longleftrightarrow B \subseteq A'$

(iv) $A_1 \subseteq A_2 \longrightarrow A_2' \subseteq A_1'$

Como se dijo antes el orden de relación de los conceptos viene dado por la generalidad del concepto, de como de preciso sea. Es decir, sean (A_1, B_1) y (A_2, B_2) dos conceptos. Si se cumple $(A_1, B_1) \leq (A_2, B_2)$, entonces esto implica que $A_1 \subseteq A_2$ y a su vez $B_2 \subseteq B_1$. Nótese que el orden de las extensiones es el orden de inclusión *inversa*, y que las propiedades antes mencionadas hacen que ambos órdenes sean, en esencia, *duales, o inversos*, el uno del otro.

Llamamos $\underline{\mathfrak{B}}(G, M, I)$ al conjunto $\mathfrak{B}(G, M, I)$ parcialmente ordenado por el orden entre conceptos anterior, $\underline{\mathfrak{B}}(G, M, I) = \langle \mathfrak{B}(G, M, I), \leq \rangle$. Pues bien, según el *teorema fundamental del Análisis en Conceptos Formales* [9], $\underline{\mathfrak{B}}(G, M, I)$ es un retículo completo en el cual ínfimo y supremo vienen dados por:

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\cap_{t \in T} A_t, (\cup_{t \in T} B_t)'' \right) \tag{1.3.3a}$$

$$\bigvee_{t \in T} (A_t, B_t) = \left((\cup_{t \in T} A_t)'', \cap_{t \in T} B_t \right) \tag{1.3.3b}$$

A veces es interesante conocer en qué concepto formal se hallan un determinado objeto o atributo. Para ello definimos dos funciones $\gamma : G \to \mathfrak{B}(G, M, I)$ y $\mu : M \to \mathfrak{B}(G, M, I)$

$$\gamma(g) = \left(\{g\}'' , \{g\}' \right) \qquad \mu(m) = \left(\{m\}' , \{n\}'' \right) \tag{1.3.4}$$

A los conceptos $\gamma(g), g \in G$ los llamamos **conceptos-objeto** y a los conceptos $\mu(m), m \in M$ **conceptos-atributo**. En este contexto, es posible demostrar que $\gamma(g)$ es una función densa bajo el operador de unión (*join-dense*) y a su vez $\mu(m)$ es una función densa bajo el operador de intersección (*meet-dense*) es decir, que el retículo se puede representar en función de supremos de conceptos-objeto e ínfimos de conceptos-atributo.

Dualmente, estas dos funciones guardan una estrecha relación con la matriz de incidencia I. Dado un retículo conceptual, podemos inducir una relación de incidencia que lo genere de la siguiente forma: si un objeto g tiene el atributo m lo escribiremos como gIm y esto es equivalente a $\gamma(g) \leq \mu(m)$ tal y como se demuestra a continuación:

$$gIm \leftrightarrow g \in m' \leftrightarrow g'' \subseteq m''' \leftrightarrow g'' \subseteq m' \leftrightarrow \gamma(g) \leq \mu(m)$$

Volviendo al ejemplo de la figura 1.2 podemos comprobar fácilmente como esa ecuación se cumple:

$$\mu(\{\text{plumas}\}) = (\{\text{avestruz, velocirraptor}\}, \{\text{bípedo, plumas, ovíparo}\}) \tag{1.3.5a}$$

$$\gamma(\{\text{avestruz}\}) = (\{\text{avestruz}\}, \{\text{ave, bípedo, plumas, ovíparo}\}) \tag{1.3.5b}$$

Mirando en la figura 1.2 podemos ver que efectivamente $\gamma(\{\text{avestruz}\}) \leq \mu(\{\text{plumas}\})$ lo que indica que el avestruz tiene plumas.

Debido a esta definición mutua en función de los polares y la relación entre conceptos-objeto y -atributo, decimos que un contexto formal y su retículo conceptual son *duales*. A veces nos referimos a esta dualidad como "el retículo es el exponencial del contexto y el contexto es el logaritmo del retículo".

Dado que cualquiera de las dos representaciones portan la misma información, nos centramos en el retículo conceptual, que recalca la relación de generalidad-especificidad entre conceptos formales, como herramienta para el análisis exploratorio.

Con el propósito de "leer" qué significa cada concepto formal en el diagrama de Hasse, se podrían anotar las extensiones e intensiones de cada concepto con un *etiquetado completo* listando para cada concepto el conjunto de objetos dentro de la extensión y de atributos de su intensión. Pero como esto implicaría repetir muchas veces cada objeto y cada atributo a través del retículo, para reducir la sobrecarga visual, se prefiere el siguiente método de *etiquetado reducido*: ponemos la etiqueta de cada atributo sólo por encima del concepto más abstracto en el que aparece, y la etiqueta de cada objeto sólo por encima del concepto menos abstracto (más específico) en el que aparece. De esta manera cada etiqueta aparece sólo una vez en el retículo reduciendo la sobrecarga visual. Esto es bastante fácil de conseguir mediante las funciones γ y μ: se anotará con la etiqueta de objeto o atributo el concepto-objeto o -atributo obtenido por estas funciones. Por ejemplo, las ecuaciones 1.3.5b y 1.3.5a muestran los conceptos en los que se hace el etiquetado.

¿Cómo se recupera, para cada concepto, su extensión e intensión completa? Para la extensión, se leen los objetos de todos los conceptos que están dominados por el concepto en cuestión, es decir hacia abajo en el diagrama de Hasse. Dualmente, para la intensión, se leen los atributos de todos los conceptos que dominan al concepto en cuestión, es decir, hacia arriba en el diagrama de Hasse.

1.3.3 La visión algebraica del Análisis en Conceptos Formales

En $\mathcal{K}$-Formal Concept Analysis $\mathcal{K}$-FCA (ver Sección 1.4) es patente desde el principio la relación de los procedimientos anteriores con el álgebra lineal. Tal conexión no es evidente a primera vista en FCA, pero es intuitiva cuando se representan los subconjuntos de un conjunto mediante el álgebra de vectores característicos: sea un subconjunto de objetos $A \subseteq G$; entonces un vector característico x_A que corresponde a dicho A está indexado por los objetos de G y $g \in A \iff x_A[g] = 1$ por lo que cualquier subconjunto $A \subseteq G$ puede ser representado como un vector de dimensión n que toma valores en $\mathbb{B} = \{0, 1\}$, $x_A \in \mathbb{B}^n$.

Entonces, por ejemplo, la relación de inclusión entre conjuntos es la relación de orden entre vectores $A_1 \subseteq A_2 \iff x_{A_1} \leq x_{A_2}$ en donde la relación de orden se considera componente a componente.

Bajo esta lectura si consideramos que $|G| = n$ y $|M| = p$ (donde las barras indican "cardinalidad" de un conjunto) entonces las extensiones son vectores indexados por objetos, las intensiones vectores por atributos y la relación de incidencia toma la forma de una una matriz de incidencia booleana $I \in \mathbb{B}^{n \times p}$, que puede ser interpretada como una transformación (lineal en un sentido analógico) entre espacios (booleanos). Es más, los polares son las *funciones*

residuales (ver el Apéndice B.4.4) por la derecha y la izquierda de esta transformación lineal y los conceptos formales pares de vectores indexados por objetos y atributos que se transforman mutuamente por residuación a través de la incidencia.

A pesar de que tratamos de glosar los detalles, esta álgebra lineal no es la estándar: vectores y matrices están definidos en un *diode*, o semianillo ordenado, (ver Apéndice B). Obviamos más explicaciones sobre esta "algebraización del FCA" porque enseguida veremos una versión mucho más productiva y compleja de ella.

1.4 Análisis en Conceptos Formales $\mathcal{K}$-valorados

En FCA un objeto puede tener o no un atributo, pero con la ampliación dada propuesta en [12] un objeto puede tener asociado un atributo en cierto *grado* que toma valor en los números reales soslayando, en parte, las limitaciones que podría traer usar sólo valores binarios. Esta extensión de FCA se llama $\mathcal{K}$-FCA, porque $\mathcal{K}$ se usa habitualmente para denotar un *semicuerpo*, que es la estructura algebraica en la que toma valores la relación entre g y m.

Análogamente al FCA, en $\mathcal{K}$-FCA la tripla $(G, M, R)_{\mathcal{K}}$ se llama el **contexto formal $\mathcal{K}$-valorado**, constituido por el conjunto de objetos G, el de atributos M un **semianillo $\mathcal{K}$ de valores** (ver Sección B.4) y una **matriz de incidencia gradual** $R \in \mathcal{K}^{n\times p}$ que relacione los objetos con los atributos de forma que $R(g, m) = \lambda$ indique que **el objeto g tiene el atributo m en grado λ** o, de forma dual, que **el atributo m es manifestado en el objeto g en un grado λ**.

Un subconjunto de objetos ($\mathcal{K}$-valorado) vendrá dado por el vector fila $x \in \mathcal{X} \cong \mathcal{K}^{1\times n}$ perteneciente a un $\mathcal{K}$-semimódulo por la izquierda (análogo a un espacio vectorial) y un subconjunto de atributos es un vector columna $y \in \mathcal{Y} \cong \mathcal{K}^{p\times 1}$ de un $\mathcal{K}$-semimódulo por la derecha[8].

1.4.1 Polares

Un resultado fundamental es que entre un conjunto de objetos G y un conjunto de atributos M se puede establecer una conexión de Galois: el $\mathcal{K}$-semimódulo por la izquierda $\mathcal{X} \cong \mathcal{K}^{1\times n}$ y el $\mathcal{K}$-semimódulo por la derecha $\mathcal{Y} \cong \mathcal{K}^{p\times 1}$ forman un *par dual* (ver B.4.6) bajo el operador $\langle \cdot | \cdot \rangle$ que se define como [12]:

$$\langle x|y\rangle_M = x^{\mathsf{t}} \otimes R \otimes y \tag{1.4.1}$$

Entre estos dos semi-módulos se establece una *conexión (contravariante) de Galois*. El **polar de extensiones** $x \in X$ se obtiene mediante la ecuación B.4.22 y despejando con ayuda de las ecuaciones B.4.10 y B.4.14:

$$(x)^{+}_{R,\varphi} = \bigvee \{\, y \in Y \mid \langle x \mid R \mid y \rangle \leq \varphi \,\} = (x^{\mathsf{t}} \otimes R) \setminus \varphi \tag{1.4.2}$$

[8] Para información más detallada acerca del semianillos y módulos ver a las secciones B.4.1-B.4.3.

El **polar de intensiones** $y \in Y$ se obtiene de forma dual, mediante la ecuación B.4.22 y despejando con ayuda de la ecuación B.4.19:

$$\overset{+}{R,\varphi}(y) = \bigvee\{x \in X \mid \langle x \mid R \mid y\rangle \leq \varphi\} = \varphi/(R \otimes y) \tag{1.4.3}$$

Esto genera el conjunto de elementos cerrados, que están fijos por la aplicación consecutiva de los polares, que hacen las veces de **extensiones e intensiones multivaloradas**:

$$\overline{\mathcal{Y}} = \{(x^{\mathrm{t}} \otimes R) \setminus \varphi | x \in \mathcal{K}^{n\times 1}\} \tag{1.4.4a}$$
$$\overline{\mathcal{X}} = \{\varphi/(R \otimes y) | y \in \mathcal{K}^{p\times 1}\} \tag{1.4.4b}$$

Tal y como se muestra en [13] se cumple la siguiente relación entre los elementos supremo- e ínfimo-irreducibles (ver B.2.4):

$$\mathcal{M}(\overline{\mathcal{X}}) = \left(\mathcal{J}(\overline{\mathcal{Y}})\right)^{-}_{\varphi} \qquad \mathcal{M}(\overline{\mathcal{Y}}) = \overset{-}{\varphi}\left(\mathcal{J}(\overline{\mathcal{X}})\right) \tag{1.4.5}$$

1.4.2 Conceptos formales y retículo conceptual

En [12, def.9] se da una definición adaptada a este nuevo marco de los conceptos formales generados por estas ecuaciones: en un semianillo reflexivo e idempotente $(\mathcal{K}, \varphi)$, en donde $\varphi \in K$, un contexto formal $\mathcal{K}$-valor $(G, M, R)_{\mathcal{K}}$ con $|G| = n$ para n finito y $|M| = p$ para p finito y los espacios vectoriales formado por las columnas $\mathcal{X} \cong \mathcal{K}^{n\times 1}$ y $\mathcal{Y} \cong \mathcal{K}^{p\times 1}$, un *$\varphi$-concepto formal del contexto* $(G, M, R)_{\mathcal{K}}$ es un par $(x, y) \in X \times Y$ tal que $x^{+}_{\varphi} = y$ y $\overset{+}{\varphi}y = x$. De esta forma x es la *extensión*, y es la *intensión* y φ su grado de existencia[9].

Se establece una relación de orden entre los elementos de forma que para los *φ-conceptos* (x_1, y_1) y (x_2, y_2):

$$(x_1, y_1) \leq (x_2, y_2) \iff x_1 \leq x_2 \iff y_1 \geq y_2$$

Al igual que para *FCA* esta relación de orden crea un conjunto parcialmente ordenado denominado $\underline{\mathfrak{B}}_{\varphi}(G, M, R)_{\mathcal{K}}$. Este conjunto parcialmente ordenado es un retículo completo llamado **retículo de φ-conceptos** de $(G, M, R)_{\mathcal{K}}$.

Además este retículo de φ-conceptos cumple la primera parte del teorema fundamental de retículos de conceptos dado por la ecuación 1.3.3. Así pues para $a_t \in \mathcal{X}$ y $b_t \in \mathcal{Y}$ se tiene:

$$\bigwedge_{t\in T}(a_t, b_t) = \left(\bigoplus_{t\in T} a_t, \left[\overset{+}{\varphi}\left[\bigoplus_{t\in T} b_t\right]\right]^{+}_{\varphi}\right) \tag{1.4.6a}$$

$$\bigvee_{t\in T}(a_t, b_t) = \left(\overset{+}{\varphi}\left[\left[\bigoplus_{t\in T} a_t\right]^{+}_{\varphi}\right], \bigoplus_{t\in T} b_t\right) \tag{1.4.6b}$$

[9] Given the algebraic flavour of the theory, we prefer to use lower case letters to denote the vectors that are (multivalued) subsets.

1.4.3 Retículos estructurales

Como el retículo $\underline{\mathfrak{B}}_{\varphi}(G, M, R)_{\mathcal{K}}$ es potencialmente infinito, con el propósito de una fácil visualización, queremos encontrar retículos que de alguna manera resuman las principales características de aquel. En [13] se propone el **retículo estructural** $\underline{\mathfrak{B}}(G, M, I_R^{\varphi})$ de un retículo de φ-conceptos $\underline{\mathfrak{B}}_{\varphi}(G, M, R)$ como el retículo conceptual del contexto booleano:

$$I_R^{\varphi}(i, j) = \left[\widetilde{\gamma}^{\varphi}(\mathbf{g}_i) \leq \widetilde{\mu}^{\varphi}(\mathbf{m}_j)\right] \tag{1.4.7}$$

Donde las funciones γ y μ se definen como se hizo en las ecuaciones 1.3.4:

$$\widetilde{\gamma}^{\varphi}(\mathbf{g}_i) = \left({}^{+}_{\varphi}((\mathbf{g}_i)^{+}_{\varphi}), {}^{+}_{\varphi}(\mathbf{g}_i)\right) \qquad \widetilde{\mu}^{\varphi}(\mathbf{m}_j) = \left({}^{+}_{\varphi}(\mathbf{m}_j), {}^{+}_{\varphi}\left((\mathbf{m}_j)^{+}_{\varphi}\right)\right) \tag{1.4.8}$$

Al igual que en la Sección 1.3 se utilizan los vectores $\mathbf{g}_i \in X$ para referirse a los objetos y $\mathbf{m}_i \in Y$ para referirse a los atributos. El conjunto de vectores indexados por los objetos $\mathbf{g}_i = [\varepsilon \cdots \epsilon_i \cdots \varepsilon]$ es una base generadora de su espacio vectorial. Lo mismo puede ser aplicado al conjunto de vectores indexados por los atributos $\mathbf{m}_i = [\varepsilon \cdots \epsilon_i \cdots \varepsilon]^T$.

Nótese que existe un retículo conceptual para cada valor del pivote φ que decidamos considerar: a la exploración de todos los diversos retículos que surgen cuando consideramos cada uno de los posibles φ lo denominaremos **exploración estructural (del contexto formal** $(G, M, R)_{\mathcal{K}}$**)**. Esta es la principal herramienta que transformaremos en un método de análisis de datos.

1.5 Paisajes del conocimiento

Uno de los autores de FCA, Wille, planteó la posibilidad de reconsiderar el conocimiento como un territorio que hay que ir descubriendo mediante exploración [14], y llamó a este marco conceptual "Paisajes del Conocimiento", del inglés Landscapes of Knowledge (LofK). Esta idea se puede enmarcar, formalizar y fundamentar en la Teoría de la Metáfora, entendida como está descrita, por ejemplo, en [15, 16].

La metáfora básica de LofK parece ser "El conocimiento no descubierto es territorio inexplorado" y hace una analogía entre la investigación dentro de campos del saber desconocidos y la exploración de terrenos sin cartografiar.

Tal y como se apunta en [17], bajo esta metáfora FCA puede plantearse como un tipo de EDA dado su énfasis en la fidelidad a los datos, importancia de la representación gráfica, su sistema de representación dual basado en diagramas de Hasse y en tablas, así como la relación entre atributos y objetos. En esta tesis, pues, usaremos "Paisajes del conocimiento", para referirnos al enunciado, desarrollo y hacer operativa esta metáfora como un método de análisis exploratorio para matrices de dos modos con valores en un semianillo idempotente.

En esta aproximación conceptual se puede dar una lista de todas las actividades de exploración del conocimiento que se encuentran apoyadas en las anteriores metáforas, y que [14] pormenorizó. A continuación las detallamos aplicadas al estudio de los datos de expresión genómica:

- **Exploración**: *Buscar algo de lo que se tiene una idea general.* Por ejemplo, se puede tener un estudio de diferentes genes en distintas condiciones y no se sabe realmente qué genes pueden presentar un cambio en función de los experimentos realizados. Una exploración en el contexto de $\mathcal{K}$-FCA al comienzo proporcionará una idea de cómo se distribuyen los datos.

- **Búsqueda**: *Buscar algo de lo que se tiene una idea más precisa pero no se puede localizar.* En el ejemplo anterior, si se tienen muestras de tejidos que se espera que tengan una respuesta similar al nivel de expresión genética, pero se desconoce qué genes podrían estar involucrados, la búsqueda de dichos genes se centrará en los niveles de expresión dados por esos tejidos.

- **Reconocimiento (o Clarificación)**: *Obtención de circunstancias y relaciones.* Por ejemplo en la Sección 5.2 se muestran diferentes tejidos de *Arabidopsis thaliana*: claramente se ve cómo genes de la planta se expresan de forma diferente en la raíz y en los brotes.

- **Identificación (o Categorización, localización en una taxonomía)** : *Examinar los datos y sus relaciones mientras se contrasta la información teórica al respecto.*

- **Investigación**:*Estudiar pormenorizadamente, examinando los resultados.*

- **Decisión**: *Resolver una situación de incertidumbre mediante un ordenamiento.*

- **Mejoras**: *Mejorar en calidad y valor.*

- **Reestructuración**: *Reformular una estructuración dada.*

- **Memorización**: *Procesar para memorizar y reproducir lo aprendido.*

Aparte de estas tareas en esta tesis se proponen dos nuevas que amplían el rango de FCA dentro de EDA:

- **Hipotetizar**: *Deducir hechos y relaciones basándose en datos.* Como por ejemplo suponiendo que los resultados de un experimento sobre los cromosomas de células pluripotenciales concuerdan con lo esperado (ver Sección 5.3).

- **Indexación**: *Indexar resultados de otras fuentes de conocimiento utilizando conceptos, objetos o atributos.* Es posible enlazar los genes con distintas bases de datos externa, así como evaluar la calidad de los conceptos formales seleccionados gracias a información obtenida de otras bases de datos (ver Sección 3.6).

Tal y como se describe en [16], el psicólogo ecologista Gibson hablaba de las propiedades que definen un objeto en el medio ambiente que permiten a un individuo realizar una acción: así, por ejemplo, un árbol será escalable. Estas interacciones proporcionadas por el ambiente son denominadas *potencialidades*(del inglés "affordance"), en palabras del propio Gibson *"propiedades de cosas tomadas como referencia a un observador, pero no propiedades de la experiencia del observador. El observador puede o no percibir o atender a la potencialidad, acorde a sus necesidades, pero la potencialidad, siendo invariante, está siempre ahí para ser percibida"* [p.216 de 16, en nuestra propia traducción].

A continuación explicaremos con más detenimiento el análisis de las potencialidades de FCA como una técnica de EDA, y como una metodología que impulsa FCA dentro de un modelo de análisis por capas orientado a suplir las necesidades del análisis de datos científicos e ingenieriles, tal y como se expone en [17]:

1. *La calidad formal de FCA lo convierte en técnica óptima para análisis de datos independiente del dominio.*
2. *Visualización y manipulación de datos en formatos de tabla y jerárquica (diagrama de Hasse) están mutuamente garantizadas.* Esto es una de las grandes fortalezas de FCA que se explicará en detalle en la Sección 3.4.2.
3. *El FCA estándar proporciona un marco genérico para trabajar de forma homogénea con tablas de dos modos de datos, con asociaciones de significado y contexto.* Esto se refiere a la relación binaria existente entre objetos y atributos que da lugar, a su vez, a contextos formales.
4. *Un retículo de conceptos describe un agrupamiento jerárquico y no partitivo entre grupos de objetos y atributos.* No partitivo significa que, aunque exista un grupo para cada conjunto de objetos y atributos, dichos grupos no son incompatibles, es decir no son, forzosamente, de intersección nula.
5. *Los sub-retículos adjuntos describen diferentes comportamientos para conjuntos asociados de objetos y atributos, cualitativa y cuantitativamente.* A veces hay subsistemas directamente observables en los datos. La identificación de sub-retículos adjuntos como fuentes independientes de estructura y análisis de sus correspondientes sub-contextos es una contribución valiosa que ayuda a la formulación de hipótesis, como por ejemplo en [18, 19].
6. *$\mathcal{K}$-FCA proporciona la exploración de datos con diferentes niveles de detalle mediante la modulación de φ.* El umbral φ de $\mathcal{K}$-FCA es una herramienta útil para explorar grados de incidencia a varios niveles de detalle, lo que se explicará con más detalle en la Sección 3.5.

1.6 Objetivos

A continuación presentamos los objetivos de investigación que nos planteamos en esta tesis, a la luz de las discusiones previas de este capítulo:

1. El Objetivo Principal de esta tesis es desarrollar un método de análisis exploratorio basado en el Análisis en Conceptos Formales $\mathcal{K}$-valorados aplicado al análisis de datos de expresión genética.

   Como sub-objetivos nos planteamos:

   (a) Ayudar al desarrollo teórico de $\mathcal{K}$-FCA como un método de exploración de datos de expresión genómica.

   (b) Construir herramientas que incorporen los desarrollos teóricos anteriores.

(c) Evaluar el desempeño de los desarrollos teóricos y prácticos anteriores.
Para este último sub-objetivo nos planteamos:
i. Evaluar el desempeño sobre datos simulados, e.g. "in silico".
ii. Evaluar el desempeño sobre datos muestreados de la realidad.

2. Como objetivo secundario nos proponemos contribuir al desarrollo de $\mathcal{K}$-FCA como un método de análisis de datos, y en particular al desarrollo de la metáfora de "Paisajes del Conocimiento".

2

El Análisis de Datos de Expresión Genética

Por la propia naturaleza y el estado de la cuestión en el estudio de la expresión genética, la mayor parte de los datos de que disponemos, aún con iniciativas como las de "open data", son no supervisados, lo que implica que muchas de las técnicas de aprendizaje y análisis predictivo no son aplicables.

En este momento del desarrollo de la cuestión, la mayor parte de las técnicas utilizables son de análisis exploratorio, lo que acarrea sus propios problemas para la evaluación de modelos y predicciones, como ocurre en el normal devenir de las Ciencias incipientes.

Dos son las técnicas mayoritariamente aplicadas en análisis de datos genéticos: el co-agrupamiento, para el desarrollo de modelos, y el análisis de enriquecimiento, para tratar de contrastar los modelos. A continuación pasamos a presentar ambos.

2.1 Co-agrupamiento

El **agrupamiento** (ing. "clustering") es una técnica de aprendizaje máquina no supervisada donde se juntan elementos (quizás bajo una misma etiqueta) en función de sus características. La entrada de un algoritmo de agrupamiento puede representarse como una matriz $M^{g \times m}$ donde se tratarán de clasificar las g filas dentro de uno o varios de los grupos detectados (ing. "clusters") basándose diferentes medidas de similitud entre vectores-fila. Uno de los más populares métodos de agrupamiento podría ser el *k-means* donde de forma iterativa se va clasificando cada uno de los g elementos dentro de su correspondiente grupo [20]. De esta forma dada una matriz los elementos se pueden clasificar por filas (o por columnas de la matriz traspuesta).

Los algoritmos de **co-agrupamiento** (ing. "coclustering; biclustering") tratan de clasificar a la vez por filas y columnas. En el caso de agrupamiento de genes interesa realizar co-agrupamiento porque se puede ver cómo el nivel de expresión de unos genes, que indexarían las filas, está afectado en ciertas muestras, que indexan las columnas, mientras parece que no se ve afectado en otras muestras. Para referirse a este tipo de algoritmos se suelen utilizar los términos en inglés *coclustering* y *biclustering*. En el ámbito de agrupamiento de genes y muestras dentro de una matriz GED se utiliza generalmente el término *biclustering*. El

término *coclustering*, aunque a veces se usa en GED, también aparecer en otros tipo de problemas como agrupar palabras y documentos [21], en publicaciones de informática. Hay que decir, como curiosidad, que el nombre fue propuesto por B. Mirkin [6] al observar el nuevo tipo de agrupamiento propuesto por Hartigan [22], que originariamente se denominó "direct clustering", o "agrupamiento directo", en español.

Por su naturaleza bimodal (ing. "two-mode data"), la literatura alrededor de co-agrupamiento de datos de expresión genética es bastante abundante. A continuación se verán diferentes tipos de algoritmos de co-agrupamiento y se realizará una descripción básica de algunos algoritmos en concreto. En la Sección 4.6 se mostrará una comparación de estos algoritmos con el método propuesto en esta tesis basado en $\mathcal{K}$-FCA.

2.1.1 Tipos de algoritmos de co-agrupamiento

Los algoritmos de co-agrupamiento se pueden dividir en diferentes tipos dependiendo del factor en el que nos fijemos. Todos los algoritmos de agrupamiento buscan maximizar (o minimizar) una función bajo diferentes criterios, y se pueden clasificar los diferentes tipos de algoritmos de co-agrupamiento basándose en dicho objetivo. Siguiendo este enfoque se podrían encontrar cinco tipos de algoritmos [23, 24]:

- métodos de minimización de la varianza.
- métodos de agrupamiento por filas y columnas.
- reconocimiento de patrones
- métodos probabilísticos y generativos.
- métodos de factores.

A continuación pasamos a describirlos brevemente.

Métodos de minimización de la varianza. Estos métodos buscan que las diferencias entre los elementos de un mismo grupo sea la menor posible. Para ello pueden utilizar diferentes medidas de distancia como la suma cuadrática media [25], la correlación [26] o la información mutua [27] entre otras.

Uno de los algoritmos más conocidos en este grupo es el algoritmo propuesto por Hartigan [22] En este algoritmo la salida está formada por un conjunto de co-grupos algunos de los cuales estarán solapados con otros, de forma que se genera una jerarquía entre co-grupos. Un co-agrupamiento $B_p = (R_p, C_p)$ está formado por las filas R_p y columnas C_p. En cada paso se divide uno de los grupos existentes en dos de forma que la suma de cuadrados se reduzca. Muchas veces la división se producirá en el punto que minimice la suma cuadrática media pero en otros casos se realizará en un punto fijado por divisiones anteriores en otros grupos. El algoritmo continúa así haciendo divisiones hasta que la calidad de los grupos obtenida mediante una división es menor que la obtenida al azar. Este algoritmo destaca por ser uno de los primeros en afrontar el problema del co-agrupamiento, por este motivo también requiere muy poca capacidad de cálculo.

Dentro de este mismo grupo, el algoritmo propuesto en [25] fue especialmente diseñado para tratar de encontrar co-agrupamientos en microarrays de expresión genética. La medida de calidad de un grupo viene dada por residuo al cuadrado medio dado por la fórmula:

$$H(I,J) = \frac{1}{|I|\,|J|} \sum_{i\in I, j\in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2 \tag{2.1.1}$$

en donde

$$a_{iJ} = \frac{1}{|J|}\sum_{j\in J} a_{ij} \qquad a_{Ij} = \frac{1}{|I|}\sum_{i\in I} a_{ij} \qquad a_{IJ} = \frac{1}{|I|\,|J|}\sum_{i\in I, j\in J} a_{ij}\,.$$

Una submatriz A_{IJ} es llamada $\delta-$*bicluster* si se cumple $H(I,J) \leq \delta$ para $\delta \geq 0$. Este algoritmo intenta conseguir n submatrices A_{IJ} con un residuo medio inferior a δ. Se comienza con una matriz A a la cual en cada iteración se van añadiendo o eliminando columnas de forma que el valor de $H(I,J)$ disminuya. Una vez se consigue $H(I,J) \leq \delta$ o el valor de $H(I,J)$ no cambia en cada nueva iteración se reporta este valor de $A(I,J)$ como un grupo. En ese momento el valor de los elementos del grupo detectado en la matriz A es enmascarado por números aleatorios, y a continuación se procede a buscar un nuevo grupo con la misma técnica iterada. Este proceso finaliza cuando se han encontrado los n grupos buscados.

Métodos de agrupamiento por filas y columnas. Se pueden aplicar algoritmos tradicionales de agrupamiento por filas y a continuación por columnas de forma iterativa tal como se hace con el Coupled Two-Way Clustering (CTWC) [28] y el sistema de visualización de grupos Furby [29].

CTWC es un algoritmo que trata de encontrar submatrices estables. Esto significa que dado un conjunto de características $\mathcal{F}_i$ y objetos $\mathcal{O}_j$, los objetos $\mathcal{O}_j$ deben ser tales que puedan ser representados usando sólo las características de $\mathcal{F}_i$. Las características $\mathcal{F}_i$ y objetos $\mathcal{O}_j$ pueden representar genes y muestras o viceversa. Pero evidentemente no se pueden encontrar todas las submatrices estables mediante fuerza bruta debido a su complejidad de cálculo. Por ello se propone utilizar un proceso iterativo.

La idea es utilizar un algoritmo de agrupamiento para independientemente agrupar genes y muestras y seleccionar los grupos que cumplen el criterio de estabilidad. El proceso iterativo comienza con la matriz completa que se divide en diferentes grupos estables de genes y muestras. Cada submatriz almacena un puntero a su matriz madre y luego, a su vez, se divide repitiendo el paso anterior. Este proceso continua hasta que ninguna de las nuevas submatrices cumpla un criterio definido.

En [28] se afirma que, aunque para sus muestras utilizan el algoritmo de agrupamiento superparamagnetic clustering algorithm (SPC), en teoría se podría utilizar cualquiera siempre que posea las siguientes propiedades:

- El número de grupos debe ser determinado por el algoritmo y no debe usarse como parámetro de entrada.
- Tiene que presentar estabilidad frente al ruido.
- Ha de generar una relación jerárquica entre subgrupos (dendrograma).

- Posee un mecanismo para identificar los subgrupos robustos y estables.
- Tiene la posibilidad de identificar grupos de forma irregular.

El algoritmo expuesto por [30] realiza también una agrupación primero en la dimensión de genes y a continuación en muestras para posteriormente combinarlos en sucesivos pasos. El algoritmo para agrupar filas o columnas de forma separada no está prescrito: puede ser k-medias o Self-organizing map (SOM) (del inglés, "Self-Organizing Map") o cualquier otro.

Reconocimiento de patrones. Este procedimiento agrupa genes que muestran un comportamiento similar, que comparten un patrón. Por ejemplo Binary inclusion-Maximal biclustering algorithm (BiMax) [31], que se explicará en detalle más adelante, o los algoritmos que utilizan técnicas espectrales como [32] que se realiza una descomposición en valores singulares. También pertenecen a esta categoría los algoritmos de tipo order-preserving submatrices (OPSMs) [33].

El algoritmo BiMax [31] es un algoritmo exhaustivo que sólo funciona con matrices binarias, donde un 1 significa que el gen i está expresado en la condición j. El algoritmo comienza dividiendo la matriz en dos C_U y C_V basándose en la primera fila como muestra la Figura 2.1. A continuación se re-ordenan las filas de forma que primero se ponen los genes que corresponden únicamente a C_U (filas G_U), después los que corresponden a C_U y C_V (filas G_W) y finalmente los que sólo corresponden a C_V (filas G_V). La submatriz con todos los elementos a 0 es descartada, y el resto de las matrices se descompone de forma recursiva. El resultado será un conjunto de submatrices anidadas.

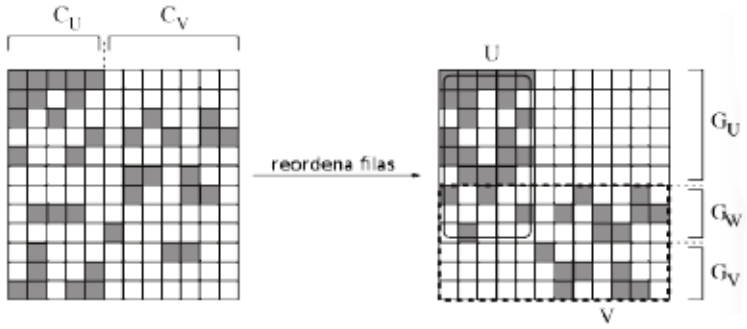


Figura 2.1: Funcionamiento de bimax, figura obtenida de [31]

Otra familia de algoritmos en este grupo sería la basada submatrices de orden preservado (OPSMs)[33]: se dice que una submatriz preserva el orden si se pueden reordenar sus columnas de tal forma que los valores de todas sus filas tengan un orden estrictamente creciente. Este requisito puede ser demasiado estricto en algunos casos de expresión genética por eso existen relajaciones. Se puede considerar que todos los genes estén ordenados pero no en el mismo sentido, unos pueden incrementar su nivel de expresión mientras otros la decrementan según cambian las condiciones. Otro ejemplo es el caso de estudios replicados: podemos tener diferentes casos de estudio y para cada uno se toman varias muestras. No tiene sentido exigir un orden específico dentro de las muestras dentro del mismo estudio, sino que se buscaría un orden entre los diferentes casos.

Sea el conjunto $T \subset 1, ..., m$ de tamaño s y $\pi = (t_1, t_2, ..., t_s)$ el orden lineal de T para formar una OPSMs. El par (T, π) es llamado el *modelo completo*. Se dice que una fila $i \in 1, ..., g$ soporta (T, π) si las correspondientes s columnas ordenadas de acuerdo con la permutación π son monótonamente crecientes. El algoritmo en cuestión tratará de encontrar un modelo completo con el soporte del máximo número de filas posible.

Encontrar el mejor modelo para un s dado se puede hacer con un algoritmo exhaustivo de búsqueda que consiste en intentar todos los modelos completos $m_s = m(m-1)...(m-s+1)$. Pero esta aproximación es muy costosa, tiene una complejidad del orden $\mathcal{O}(gm^{s+1})$, por ello se utiliza una técnica más sencilla que consiste en encontrar *modelos parciales*. Un modelo parcial (a, b) especifica los índices de los a más pequeños elementos $< t_1, .., t_a >$ y los b más grandes elementos $< t_{s-b+1}, ..., t_s >$ de un modelo completo (T, π). Sea $\theta = < t_1, ..., t_a >, < t_{s-b+1}, ..., t_s >, s$ un modelo parcial. Si $a + b < s$ entonces θ puede ser extendido a varios modelos completos. Empezando por todos los $m(m-1)$ modelos parciales de orden $(1, 1)$, se elije los mejores ℓ modelos parciales, y para cada uno de ellos se intentan todas las $m-2$ extensiones para el modelo parcial $(2, 1)$. De nuevo se elijen los ℓ mejores modelos parciales y se intenta con las $m-3$ extensiones que quedan de modelos parciales de orden $(2, 2)$. Este proceso continua hasta que se tienen $(\lceil s/2 \rceil, \lfloor s/2 \rfloor)$. Estos serán modelos completos, así que la salida es el de mejor calidad.

Modelos probabilísticos y generativos. Estos algoritmos asumen ciertos modelos para definir los co-agrupamientos. Por ejemplo pueden asumir que la matriz de expresión genética puede ser descompuesta en diferentes modelos probabilísticos o que la relación entre genes y muestras se puede interpretar como un grafo bipartito. Estos algoritmos generalmente funcionan bien siempre y cuando esas asunciones sean ciertas.

Por ejemplo los modelos cuadriculados (ing. "plaid models") [34, 35] asumen que la matriz GED está formada por diferentes capas, de cuyo resultado solo vemos la suma total. Así el valor de un elemento de la matriz A vendría dado por la fórmula:

$$a_{ij} = \mu_0 + \sum_{k=1}^{K} (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} \kappa_{jk} + \epsilon_{ij} \tag{2.1.2}$$

Donde $\rho_{ik} \in 0, 1$ y $\kappa_{jk} \in 0, 1$ valen 1 si el gen i para la muestra j pertenecen al grupo k; en caso contrario su valor es 0. Las variables α_{ik} y β_{jk} indican respectivamente el nivel medio de cada gen y columna en la capa k. El valor μ_k simplemente es el valor de fondo de la capa k. Y por último ϵ_{ij} indica el error entre el valor real de expresión a_{ij} y el estimado por el modelo de capas. Este algoritmo trata de minimizar el valor de ϵ_{ij} utilizando mínimos cuadrados: para ello también elimina la restricción binaria de ρ_{ik} y κ_{jk} y permite que su valor varíe en el rango $[0, 1]$ hasta que se obtiene una solución.

Otro tipo de algoritmos asumen que la relación entre los genes y las muestras dada por la GED puede verse como un grafo bipartito donde los vértices están formados por el conjunto de genes y muestras y ambos están relacionados mediante aristas con un valor proporcional al indicado en la GED. Muchas de estas técnicas se centran en minimizar el número de aristas entre grupos[36, 37, 38]. Finalmente, es bien conocido en la comunidad FCA que un grafo bipartito es un "criptomórfico" de un contexto formal, es decir, existe un isomorfismo no obvio entre ambos [39].

Métodos de factores. Este tipo de algoritmos como Factor analysis for bicluster acquisition (FABIA)[23] tienen en cuenta las dependencias lineales entre los niveles de expresión genética y las condiciones. Esto ayuda a capturar las distribuciones de cola pesada, que son en teoría las que aparecen con mayor frecuencia en las GED.

El algoritmo FABIA utiliza factores lo que significa que dos vectores se consideran equivalentes si el ángulo que se forma entre ellos es 0º o 180º, o dicho de otra forma el valor absoluto de su correlación es 1. Se permite intersecciones entre grupos pero ningún grupo debe estar completamente contenido dentro de otro. Esto marca una diferencia importante con $\mathcal{K}$-FCA donde sí se permite que haya grupos contenidos dentro de otros.

Suponiendo que tenemos k grupos, la matriz GED vendrá dada por la fórmula:

$$\mathbf{X} = \sum_{i=1}^{k} \lambda_i z_i + \Upsilon = \Lambda Z + \Upsilon \tag{2.1.3}$$

Donde $\lambda_i \in \mathbb{R}^{g\times 1}$ es un vector columna con el valor 0 para los genes que no pertenecen al co-agrupamiento i y otro valor para el resto. A su vez el vector fila $z_i \in \mathbb{R}^{1\times m}$ definirá las columnas de GED que pertenecen al co-agrupamiento, los elementos con valor 0 indican que esa muestra no pertenece al co-agrupamiento i. La matriz $\Upsilon \in \mathbb{R}^{g\times m}$ modela el ruido aditivo. Esta representación guarda cierta similitud con los algoritmos tipo *plaid* dados por la fórmula (2.1.2). Esta misma expresión se puede representar como el producto de dos matrices, por un lado $\Lambda \in \mathbb{R}^{g\times k}$ contiene todos los vectores λ_i y por otro $Z \in \mathbb{R}^{k\times m}$ contiene a los vectores z_i, coindexados con los anteriores. Un ejemplo de cómo aparecen estos co-agrupamientos se muestra en la Figura 2.2.

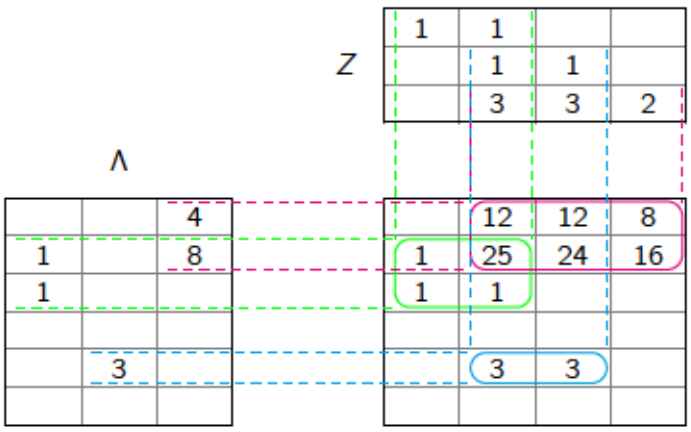


Figura 2.2: Ejemplo de co-agrupamiento generado por FABIA. La matriz de expresión genética M está formada por la expresión de 6 genes para 4 muestras, en donde se encuentran 3 co-agrupamientos diferentes. La matriz de la izquierda (Λ) muestra los grupos de genes, cada fila representa un gen y cada columna un grupo diferente. La matriz Z muestra los grupos de muestras: cada columna representa una muestra y cada fila un grupo diferente. El resultado de multiplicar $\Lambda \times Z$ da como resultado una matriz similar a la matriz de expresión genética pero sin ruido, sólo con el nivel de expresión de los genes.

La ecuación 2.1.3 se puede interpretar como un análisis de factores donde $\mathbf{X}$ es la matriz de observación, Λ es la matriz de carga ("load matrix" en inglés), y Z es la matriz de factores

en la que la columna j es el vector de factores al que contribuye la muestra j. Las matrices de carga y de factores se modelarán siguiendo una distribución de Laplace como se describe en [23].

Los valores de las matrices se obtienen mediante el algoritmo de maximización de la esperanza [40] ("Expectation Maximization (EM)", por sus siglas en inglés). Los co-agrupamientos entonces son ordenados en función de la información mutua existente entre X y Z. En este momento se tiene una dependencia blanda entre los grupos y las muestras y genes, es decir, un gen y muestra dependen en cierto grado a un grupo. Para crear co-agrupamientos bien definidos se establece un umbral en función de la media y varianza que determinará si un elemento de la matriz X pertenece a un co-agrupamiento o no.

2.1.2 Tipos de co-agrupamientos

Existen diferentes formas de generar los agrupamientos atendiendo a diversas características de su estructura en función de la jerarquía y del solapamiento que se permita entre ellos. Como se dice en [41], básicamente existen cuatro tipos de agrupamientos por filas o columnas: aquellos cuyas filas/columnas no se solapan, los que sí se solapan y los grupos que se encuentran anidados unos dentro de otros. Se pueden hacer varias combinaciones entre ellos en función de si, por ejemplo, se permite solapamiento entre las filas pero no en las columnas, o se anidan los grupos sólo por columnas y no por filas. En [24] se desarrolla un poco más esta idea y propone varios ejemplos que se muestran en la Figura 2.3.

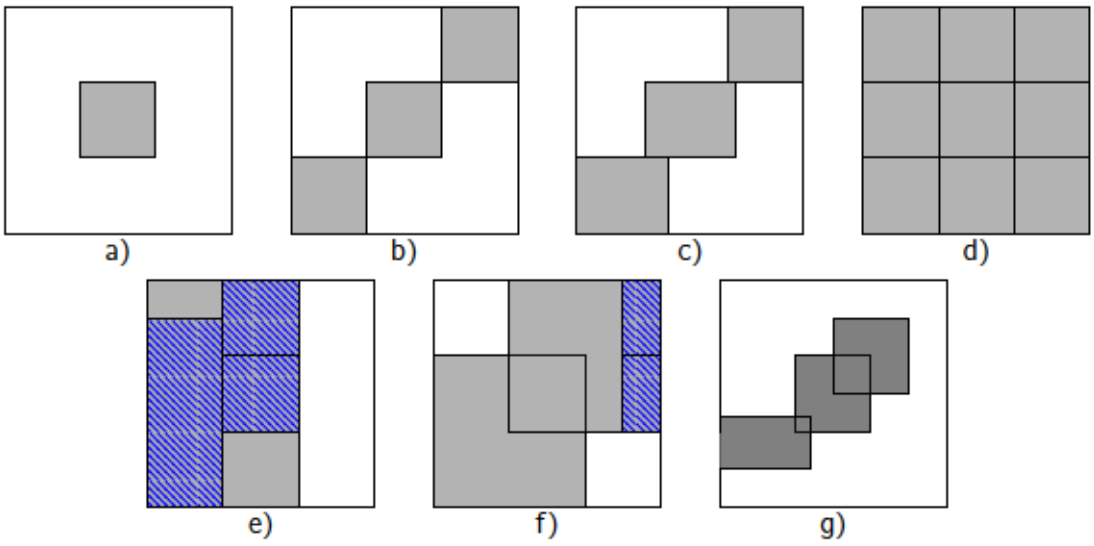


Figura 2.3: Diferentes tipos de agrupamientos que pueden ser detectados en función del algoritmo. a) Un sólo co-agrupamiento. b) Agrupamientos exclusivos por filas y columnas. d) Agrupamiento exclusivo por filas. d) Grupos no solapados formando una cuadrícula. e) Grupos no solapados con jerarquía. f) Grupos solapados con jerarquía. g) Grupos arbitrarios con solape.

Co-agrupamiento sencillo. En el co-agrupamiento sencillo, sólo un grupo es detectado por cada iteración del algoritmo. Evidentemente, sucesivas iteraciones del algoritmo podrían

generar otros co-agrupamientos diferentes. El algoritmo propuesto por [25] se encarga de encontrar una submatriz que cumpla con los requisitos del grupo y a continuación sustituye los elementos de esta submatriz por otros valores aleatorios en la matriz principal.

Agrupamiento exclusivo por filas y columnas. Las filas y columnas de estas matrices pertenecen en exclusiva a un grupo. Ejemplos de este tipo de clasificación se pueden encontrar en [37, 38] donde convierten la matriz GED en un grafo y tratan de minimizar los cruces de aristas. Generalmente no son buenos algoritmos preparados para analizar matrices de expresión genética, que no parecen exhibir agrupamientos de ese estilo.

Agrupamientos exclusivo por filas o columnas. Únicamente las filas de estas matrices pertenecen en exclusiva a un grupo. De forma dual, estos algoritmos pueden suponer que las columnas de las matrices pertenecen en exclusiva a un grupo como por ejemplo el propuesto en [30] y explicado en la sección anterior.

Cuadrícula. En este caso los grupos no se solapan y no existe ningún tipo de jerarquía entre ellos, y todos los elementos de la matriz GED pertenecen a un grupo. Existen multitud de ejemplos en este tipo de algoritmos. Por ejemplo la propuesta de [32] que se basa en la descomposición espectral de la matriz GED.

En Crowding distance based Multi-objective Particle Swarm Optimization Biclustering (CMOPSOB) [42], sin embargo, se utilizan algoritmos basados en optimización por enjambre para crear submatrices cuyos genes comparten unos niveles de correlación altos para ciertas condiciones. También pertenece a este grupo el algoritmo Microarray data CLustering Using Binary Splitting (M-CLUBS) [43] donde se trata de minimizar la distancia cuadrática media entre los elementos de un mismo grupo.

Grupos no solapados con jerarquía. Los grupos de genes se pueden solapar entre sí sólo si todos los elementos de un grupo están contenidos dentro de otro. Esto se podría considerar como una especie de jerarquía, como la propuesta por Hartigan originalmente [22]. También el algoritmo CTWC [28] cumple estas características.

Otro ejemplo de este tipo de algoritmos se puede encontrar en [44], donde definen un modelo basado en distribuciones binomiales negativas para tratar de estimar el comportamiento de un análisis RNA-Seq. Esto genera una serie de co-agrupamientos en estructura de árbol que no se solapan entre si.

Grupos solapados con jerarquía. Estos algoritmos muestran una jerarquía entre grupos que además pueden solaparse entre sí. Los algoritmos basados en FCA cumplen esta premisa.

Grupos con solape arbitrario. En este tipo de algoritmos no hay ninguna limitación en cuanto al solape que pueden tener los co-agrupamientos. Tampoco existe una jerarquía entre los distintos grupos. El algoritmo Cheng&Church[25] tras varias iteraciones corresponde a este grupo. También pertenecen a él los algoritmos de modelo cuadriculado [34].

2.2 Análisis de enriquecimiento de datos

Se conoce por *análisis de enriquecimiento de genes* (GEA, del. ing. "gene enrichment analysis") al conjunto de técnicas que permiten evaluar la función biológica de un grupo de genes. Mediante el uso de información externa se puede calcular la significación estadística de que un grupo de genes dado se haya clasificado bajo la misma etiqueta [45].

Dentro de GEA el análisis de sobre-expresión, que es el que nos interesa en este caso, consiste en obtener una lista de genes candidatos, y otra una lista previamente conocida de genes que pertenecen a una categoría dada por algún sistema externo. Mediante el uso de una distribución hipergeométrica se calcula el p-valor del conjunto de genes candidatos [46].

Esta idea se explota en [47] para clasificar los genes de cada grupo en una de las 199 categorías funcionales dentro de la base de datos de Martinsried Institute of Protein Sciences (MIPS) para la levadura. Con esta información calcula la probabilidad de que varios genes dentro de cierta categoría funcional hayan acabado juntos en el mismo grupo por azar. En [48] se realiza el mismo ejercicio pero utilizando la base de datos de GO, esta enorme base de datos clasifica en diferentes ontologías miles de genes de varios organismos[49].

También existen diferentes herramientas que permiten aprovechar la información de GO a la hora de analizar grupos de genes sobre-expresados que se pueden consultar en `http://geneontology.org/page/go-enrichment-analysis`.

2.3 Uso de Análisis en Conceptos Formales en GED

Es evidente que el FCA estándar está relacionado con el co-agrupamiento: los conceptos formales son agrupamientos de objetos y atributos mutuamente definidos [7]. Pero a pesar de este prometedor punto de partida, la relación entre el co-agrupamiento y FCA no ha sido explorada satisfactoriamente, y en particular, en lo que atañe al análisis de datos de expresión genética. Para otros análisis no supervisados la mejor fuente sigue siendo [50].

Pensa introduce en [51] un sistema para analizar las matrices GED de una forma parecida a la técnica propuesta en esta tesis. En este caso se trabaja con la traspuesta de la matriz GED donde las filas representan la condición biológica a la que se ha sometido la muestra y cada columna muestra el nivel de expresión de cada gen. A partir de este punto define un umbral en función del valor de expresión de cada gen que será utilizado para convertir la matriz de expresión en una booleana. También se introduce la idea de *contexto enriquecido*: a la matriz booleana generada tras pasar la GED por el umbral, se le añaden nuevas columnas que definirán condiciones o atributos conocidos a priori acerca de las muestras. Por ejemplo si se tiene un estudio sobre el cáncer con muestras que pertenecen a tejidos sanos o cancerígenos se podría indicar con una columna cuyos elementos indiquen si es un tejido canceroso o no. Aunque podría pensarse que esta manera de proceder proporciona determinado grado de supervisión a los contextos formales, hay que recordar que, en las técnicas de construcción de retículos, no tiene cabida dicha supervisión.

Por otra parte en [52, 53] se habla de estructuras de patrones salidas de diferentes intervalos. Esto es, la matriz booleana a partir de la GED se consigue mediante umbrales

como en el caso anterior, pero en este caso se utilizan varios umbrales diferentes. De esta forma se analizan intervalos de forma simultánea, no un sólo valor.

Existen, sin embargo, varios ejemplos de uso de FCA en temas relacionados con GED, no sólo limitados a co-agrupamiento, como podrían ser los casos anteriores, sino también para la identificación de biomarcadores, como hacen en [54], que posteriormente se utilizarán para indicar si un tejido tiene cáncer o no. También se puede usar FCA para tratar de detectar relaciones desconocidas en un red regulatoria de genes como se propone en [55].

En una ambiciosa panorámica histórica de los esfuerzos rusos por usar retículos para la minería de datos y el aprendizaje máquina figura notoriamente el FCA [56]. En particular, el autor propone una teoría de aprendizaje inductivo que equipara a los árboles de clasificación, pero en ningún momento evalúa. Esta deriva también se observa en [57], y es la aproximación que se usa en [52, 53] para tratar GED mencionados arriba.

Para terminar, parece ser que el ámbito de aplicación más productiva en la actualidad de los retículos conceptuales es como una herramienta de exploración del conocimiento [58, 59, 60, 61], lo que se puede aplicar, en particular, a la exploración bibliográfica, como en las referencias mencionadas.

Existen dos dificultades técnicas principales en estos modelos: el carácter no supervisado de los algoritmos de inducción de retículos conceptuales estándares y la fragilidad frente a la presencia de ruido booleano de tipo "sal y pimienta" en la matriz de incidencia.

Con respecto al primero de estos asuntos, existen algunas propuestas para construir clasificadores, es decir, sistemas con supervisión basados en FCA, como algunos de los recogidos en [56], pero no conocemos evaluaciones fidedignas de sus capacidades en lo relativo a la clasificación o a la predicción: probablemente estamos en un estadio demasiado temprano del desarrollo de la tecnología, aunque también es posible que FCA sea "esencialmente" una tecnología para problemas no supervisados, como la inducción de reglas (con la que FCA tiene otro criptomorfismo) o el (co-)agrupamiento.

Con respecto al segundo, problema en el manual de FCA [7], ya se hace referencia al hecho de que "desmantelar" una relación mediante la retirada de un bit o su adición implica cambios notables en la composición del retículo conceptual en términos de conceptos formales que aparecen o desaparecen. Este comportamiento, que parece un ejemplo de discontinuidad inherente de la técnica, sugirió la creación de alternativas no booleanas, y en particular de $\mathcal{K}$-FCA [12, 62, 63] (y también la Sección 1.4).

El modelo que desarrollamos en esta tesis se ha presentado parcialmente en [17, 64, 65] y presenta la ventaja frente a los anteriores de que está concebido y desarrollado como un método de análisis exploratorio, no confirmatorio. En ese aspecto está preparado para interaccionar con otros métodos como el propio FCA, bajo su consideración como un método de co-agrupamiento, y con otras fuentes de datos.

En particular, la asunción de que un retículo conceptual es también un mecanismo de indexación de información permite considerar los reticulos conceptuales en relación a las ontologías de genes para usos de análisis de enriquecimiento, GEA. En la Sección 3.6.2 nos centraremos en el desarrollo de esta técnica, que se aplicará al análisis $\mathcal{K}$-FCA para evaluar la calidad de los grupos generados.

Además, como una herramienta adicional de visualización se muestra la posibilidad de ver la relación existente entre términos GO y genes en un retículo de conceptos, algo que puede aportar información adicional a la relación que existe entre genes de un mismo grupo. Esto se explicará en la Sección 3.7.

3

Análisis de Datos de Expresión Genética con Análisis en Conceptos Formales $\mathcal{K}$-Valorados

3.1 Introducción

El análisis de expresión genética parte de una matriz de expresión de datos que muestra el nivel de expresión de miles de genes en diferentes muestras de células, cada una de las cuales podría estar sometida a diferentes condiciones biológicas como ya se adelantó en la Sección 1.1.

Continuando con las metáforas desarrolladas en torno a FCA expuestas en la Sección 1.5 se puede pensar en un gen como en un objeto que, a su vez, puede tener diferentes atributos, cada uno correspondiente a una de las muestras de células. En esta aproximación que hacemos, pues, las filas de la matriz están indexadas por los genes y las columnas por las muestras (o condiciones de expresión).

El *concepto formal* estará formado por un conjunto de genes que se encuentran sobre-expresados (o infra-expresados si se hace el $\mathcal{K}$-FCA en el semicuerpo inverso) en diferentes muestras. El proceso de análisis exploratorio que se propone en esta tesis para finalmente obtener el conocimiento derivado de los *conceptos formales* se puede dividir en varios pasos [17]:

1. **Contextualización**. Primero es necesario entender qué es lo que se desea obtener, qué se quiere comparar y qué datos será necesario estudiar. En función del tipo de datos que se tengan habrá que realizar algún proceso para extraer la matriz de GED. Por ejemplo si los datos de que se dispone provienen de un fichero CEL habrá que realizar un procesado de eliminación del ruido de fondo como el explicado en el Apéndice A.5.1. Si por el contrario los datos tratan de cadenas de RNA secuenciadas, habrá que realizar el conteo de genes como se explica en el Apéndice A.5.2. La contextualización define los límites del conocimiento que se puede observar, ya que permite concentrarse en un tipo de información en detrimento de otra.

2. **Preparación de datos**. El interés de los datos se basa en cómo cambian los genes su nivel de expresión relativo en función de la diferentes condiciones de expresión que acontecen en cada muestra. Para ello hace falta realizar un pre-procesado como se detalla en la Sección 3.2.

3. **Exploración del número de conceptos**. Una característica importante de $\mathcal{K}$-FCA es la evolución del número de conceptos a lo largo de los umbrales φ y ϕ dentro de la exploración maxplus y min-plus. Esta parte se detallará en la Sección 3.3.

4. **Exploración del coagrupamiento de genes**. Mediante las técnicas de visualización propuestas en las Secciones 3.4 y 3.5 se pueden identificar los grupos de genes y muestras de interés.

5. **Análisis**. Una vez identificados los grupos de interés se puede realizar un análisis más exhaustivo con ayuda de bases de datos externas como se explica en las Secciones 3.6 y 3.7. Se puede conocer qué ontologías comparten los genes para tratar de caracterizar el grupo y ponerlo en contexto.

Todo este proceso se enmarca dentro de lo que se podría considerar como EDA porque no realizamos ningún tipo de suposición respecto a los datos de entrada. Además el análisis se va enfocando hacia un lugar u otro en función de los resultados que se van obteniendo, de una forma similar a como se exploraría un terreno desconocido con un mapa, de ahí la metáfora de LofK a la se hace referencia toda la Sección 1.5. El análisis a continuación descrito trata de proporcionar una guía a seguir para extraer la información de forma visual a través de diferentes retículos de conceptos y contrastarla con los datos conocidos de bases de datos externas.

3.2 Pre-procesado

El pre-procesado consiste en adaptar los datos que se obtienen directamente de la matriz de expresión genética para prepararlos como entrada al análisis. Como el nivel de expresión de los genes no es comparable entre sí, cada gen puede mostrar un nivel de expresión diferente dependiendo de su naturaleza. Más que la cantidad absoluta cuantitativa de un gen, interesa averiguar cómo cambia su nivel de expresión, comparándolo en las diferentes condiciones biológicas a la que se ha visto sometido en diferentes muestras. Por ello, para hacerlos comparables, hará falta realizar un pre-procesado en el que podemos distinguir dos fases, la primera consistente en la adaptación de la matriz a las necesidades del estudio basándose en conocimiento *a priori*, y la segunda consistente en una normalización para adaptarla al rango de valores de entrada del algoritmo.

Hay que tener en cuenta que cuando se obtiene la matriz GED ya se ha realizado un paso previo de pre-procesado donde el nivel de expresión de cada muestra se ha normalizado para que presenten unos valores comparables. Esto con los microarrays se consigue con el algoritmo robust multi-array analysis (RMA)[66] entre otros. Para el caso de análisis RNA-Seq se supondrá que otro tipo de pre-procesado similar se ha utilizado. Así pues la matriz GED que se recibirá como parámetro de entrada de este algoritmo en teoría no debería necesitar ningún tipo de normalización por columnas (muestras).

3.2.1 Adaptación de GED

En diferentes ocasiones se tiene conocimiento *a priori* de los experimentos que se realizan, lo cual puede contribuir a simplificar la matriz de expresión genética. Esta adaptación puede dividirse en dos casos, por filas o por columnas.

3.2.1.1 Adaptación por filas

En el caso de GED obtenidas del estudio de microarrays, lo que se obtiene es el nivel de expresión no del gen sino de una **sonda** (del inglés, "probeset"). Una sonda busca pequeñas cadenas de ARNm y cuantifica su concentración. Para cada gen hay varias sondas y hay genes que comparten sondas. Una descripción más detallada entre la relación de genes y sondas se puede leer en el Apéndice A.5.1. Combinando el valor de varias sondas que pertenecen al mismo gen es posible obtener el nivel de expresión de un gen. Para ello existen diferentes alternativas [66, 67]:

- **Promedio**. Calcula la media aritmética del nivel de expresión de todas las sondas que pertenecen a un mismo gen.
- **Aleatorio**. Selecciona de forma aleatoria una sonda entre todas las pertenecientes al gen para representar el nivel de expresión.
- **Media**. Se selecciona la sonda de mayor valor medio a lo largo de las muestras.
- **Varianza**. Se selecciona la sonda de mayor varianza a lo largo de las muestras.
- **Correlación**. Calcula la correlación de una sonda en los distintos experimentos respecto a las otras sondas del mismo gen, suma el valor de las correlaciones para cada sonda y selecciona el valor de la sonda que presenta un mayor valor en la suma de correlaciones.
- **Entropía**. Se elije la sonda con un mayor valor de entropía, definida como $H(x) = -\sum_i f_m(x_i) \cdot log(f_m(x_i))$ donde $f_m(x_i)$ es el nivel de expresión porcentual de la sonda x_i.

3.2.1.2 Adaptación por columnas

Como se explica en la Sección 3.4.2 el número de conceptos máximo es 2^m donde m es el número de columnas que tiene la matriz de expresión genética. Como aumentar en uno el número de experimentos multiplica potencialmente por 2 el total de conceptos que pueden aparecer en el retículo, se ve que este crecimiento exponencial rápidamente hace que no se puedan analizar directamente con la tecnología actual matrices de expresión con muchas (aprox. más de 10) columnas.

Muchas veces se toman varias muestras de un mismo experimento para reducir los efectos del ruido durante el análisis. Así, diferentes muestras que corresponden con un mismo grupo se pueden combinar, y la forma más común para combinar varias columnas en una es tomar la media. De esta forma se pueden analizar experimentos que tienen decenas de muestras: al combinar columnas se reduce el número de conceptos que pueden aparecer, y, además, se robustecen los estimadores de nivel de expresión.

3.2.2 Normalización

El interés de este tipo de experimentos se centra en ver el ratio del nivel de expresión de un gen en diferentes muestras, y múltiples estudios [23, 68, 69, 70, 71] consideran agrupar genes basándose en dicho ratio en función del experimento (mayor que uno o menor que uno, en inglés *n-fold variation*). Por este motivo en varias normalizaciones propuestas se aplica un logaritmo para transformar los cambios multiplicativos en aditivos.

En esta tesis se han estudiado diferentes métodos de normalización que se explican con detalle en los siguientes apartados. La matriz de expresión genética vendrá dada por $A \in \mathcal{R}_{\geq 0}^{g \times m}$ y el resultado después del pre-procesado será $R \in \mathcal{R}^{g \times m}$, listo para un análisis de expresión explicado en la Sección 3.3.

3.2.2.1 Normalización respecto a la media

Esta normalización consiste en calcular la media del nivel de expresión de cada gen en los experimentos a estudiar y después dividir el valor de cada elemento de la matriz GED por dicha media. A continuación se aplica el logaritmo. Así los valores estarán en el rango $(-\infty, \infty)$ con valores cercanos a 0 cuando el nivel de expresión del gen apenas varíe entre las diferentes muestras, los valores más negativos indicarán un nivel de infra-expresión menor que el de la media de experimentos, un nivel de expresión positivo indica lo contrario.

Existen dos variantes de este pre-procesado en función de la media deseada: hemos probado a utilizar la media aritmética:

$$r_{ij} = log\left(\frac{a_{ij}}{\frac{1}{m}\sum_{k=0}^{m-1} a_{ik}}\right) \tag{3.2.1}$$

Y también la media geométrica:

$$r_{ij} = log\left(\frac{a_{ij}}{\sqrt[p]{\prod_{k=0}^{m-1} a_{ik}}}\right) \tag{3.2.2}$$

3.2.2.2 Normalización por filas a media 0 y varianza 1

El objetivo es tener una matriz con filas cuya media aritmética sea 0 y varianza 1 [72, 73]. Para ello se resta a cada uno de los elementos de una fila la media aritmética de su fila y se divide por la desviación típica de la fila. La fórmula resultante es:

$$r_{ij} = \frac{a_{ij} - \overline{a_{i\cdot}}}{\sqrt{\frac{1}{m}\sum_{k=0}^{m-1}\left(a_{kj} - \overline{a_{i\cdot}}\right)^2}} \tag{3.2.3}$$

También se puede usar a priori el logaritmo de la matriz A para obtener una resultado similar:

$$r_{ij} = \frac{log(a_{ij}) - \overline{log(a_{i\cdot})}}{\sqrt{\frac{1}{m}\sum_{k=0}^{m-1}\left(log(a_{kj}) - \overline{log(a_{i\cdot})}\right)^2}} \tag{3.2.4}$$

3.2.2.3 Normalización por filas y columnas

Otro sistema de normalización es el descrito en [74] que propone realizar una normalización simultánea por filas y columnas donde se intenta conseguir que su media sea 0 y varianza 1. La normalización se realiza de forma similar a la indicada en la Sección 3.2.2.2, pero de forma alternativa e iterativa en filas y columnas hasta que converja a una matriz con filas y columnas con media 0 y varianza 1.

El problema de realizar la normalización por columnas en experimentos con microarrays es que cuando se genera la matriz GED a partir de los ficheros CEL ya se realiza una normalización por columnas. Una nueva normalización en este caso sería contraproducente.

3.3 Análisis de la infra- y sobre-expresión genética

Como se ha comentado con anterioridad lo que se va a tratar de identificar son los genes que presentan un cambio del nivel de expresión en diferentes muestras, lo que interesa es la variación del nivel de expresión no su valor absoluto. Así pues consideramos que:

- un gen está **infra-expresado** bajo una determinada condición cuando su nivel de expresión es inferior que bajo otras condiciones, y
- un gen está **sobre-expresado** en caso contrario, cuando su nivel de expresión sea mayor respecto a otras condiciones.

Como esta definición es relativa a un nivel de expresión considerado "normal", a menudo hay muestras de lo que se supone que es el "estado normal de un espécimen".

3.3.1 Fundamentos genéricos

El $\mathcal{K}$-FCA permite modular qué tipo de exploración se realiza cambiando el álgebra $\mathcal{K}$, que prescribe cómo cuantificar los datos y cómo operar con sus valores. Consecuentemente, para identificar cuáles son los genes sobre-expresados y cuáles los infra-expresados en $\mathcal{K}$-FCA cambiamos el álgebra en el que se ejecuta el procedimiento de exploración de la matriz de expresión:

- para estudiar la infra-expresión se usa el **semianillo max-plus** [75]

  $$\mathbb{R}_{\text{máx},+} = \langle \mathbb{R} \cup \{-\infty\}, \vee, +, -\infty, 0 \rangle$$

  que es un semianillo idempotente donde la adición es el máximo de dos valores $a \oplus b = a \vee b$ y el producto es la suma $a \otimes b = a + b$. El elemento neutral para la adición y el producto son $-\infty$ y 0 respectivamente.

- en sentido contrario, para estudiar la sobre-expresión se usa el **semianillo min-plus** [76]

  $$\mathbb{R}_{\text{mín},+} = \langle \mathbb{R} \cup \{+\infty\}, \wedge, +, +\infty, 0 \rangle$$

  que tiene definida la adición como el mínimo de dos valores $a \oplus b = a \wedge b$. Su elemento neutral para la adición es $+\infty$.

Más información acerca de semianillos y sus propiedades se puede encontrar en el Apéndice B.4.2.

Existe una pequeña dificultad técnica con estos dos semianillos, que es que $\mathcal{K}$-FCA necesita semicuerpos completos, pero estos dos semianillos no lo son porque su ínfimo ($\bot$) no tiene inverso. Esta cuestión puede ser resuelta aplicando [77, construction 2] donde el producto se extiende a dos operaciones diferentes $\dot{\otimes}$ y $\underset{\cdot}{\otimes}$:

$$a \underset{\cdot}{\otimes} b = a \underset{\cdot}{+} b = \begin{cases} -\infty & \text{if } a,b \in \mathbb{R} \cup \{\pm\infty\}, \text{ with } a = -\infty \text{ o } b = -\infty \\ +\infty & \text{if } a,b \in \mathbb{R} \cup \{+\infty\}, \text{ with } a = +\infty \text{ o } b = +\infty \\ a+b & \text{if } a,b \in \mathbb{R} \end{cases} \tag{3.3.1a}$$

$$a \dot{\otimes} b = a \dot{+} b = \begin{cases} +\infty & \text{if } a,b \in \mathbb{R} \cup \{\pm\infty\}, \text{ with } a = -\infty \text{ o } b = -\infty \\ -\infty & \text{if } a,b \in \mathbb{R} \cup \{-\infty\}, \text{ with } a = -\infty \text{ o } b = -\infty \\ a+b & \text{if } a,b \in \mathbb{R} \end{cases} \tag{3.3.1b}$$

También se cumple la siguiente ecuación [78] $a \dot{\oplus} b = \left(a^{-1} \underset{\cdot}{\oplus} b^{-1}\right)^{-1}$ que en el semianillo $\mathbb{R}_{\text{máx},+}$ quedaría como $a \wedge b = \left(a^{-1} \vee b^{-1}\right)^{-1}$. En esta última formula, el inverso multiplicativo maxplus es $a^{-1} = -a$.

El semianillo max-plus completo es $\overline{\mathbb{R}}_{\text{máx},+} = \left\langle \mathbb{R} \cup \{\pm\infty\}, \vee, \dot{+}, -\infty, 0 \right\rangle$ y el semianillo min-plus completo es, de forma dual, $\overline{\mathbb{R}}_{\text{mín},+} = \left\langle \mathbb{R} \cup \{\pm\infty\}, \wedge, \dot{+}, \infty, 0 \right\rangle$. Con la inversión mencionada en el párrafo anterior, ambos son semicuerpos completos.

Para el semimódulo de matrices finitas explicado en el Apéndice B.4.3 se definen las siguientes operaciones de multiplicación de matrices para $\mathbb{R}_{\text{máx},+}$ y $\mathbb{R}_{\text{mín},+}$:

$$\left(A \dot{\otimes} B\right)_{ij} = \sum_{k=1}^{n}{}^{\bullet} (A_{ik} \dot{+} B_{kj}) = \bigvee_{k=1}^{n} \left(A_{ik} \dot{+} B_{kj}\right) \tag{3.3.2a}$$

$$\left(A \underset{\cdot}{\otimes} B\right)_{ij} = \sum_{k=1}^{n}{}^{\bullet} (A_{ik} \underset{\cdot}{+} B_{kj}) = \bigwedge_{k=1}^{n} \left(A_{ik} \underset{\cdot}{+} B_{kj}\right) \tag{3.3.2b}$$

3.3.2 Análisis de la infra-expresión con el semicuerpo max-plus

Mediante el semicuerpo max-plus $\overline{\mathbb{R}}_{\text{máx},+} = \langle \overline{\mathbb{R}}, \vee, \dot{+}, -\infty, 0, \infty \rangle$ se van a poder identificar los grupos de genes infra-expresados, aquellos que presentarán un nivel de expresión relativo inferior a una condición de referencia. Aunque el procedimiento genérico de exploración del $\mathcal{K}$-FCA fue descrito en la Sección 1.4 de la Introducción, a continuación lo particularizaremos usando el semicuerpo max-plus.

Los polares de $\mathcal{K}$-FCA se generan a partir de la residuación del producto escalar $\langle x \mid R \mid y \rangle = x^{\mathsf{t}} \underset{\cdot}{\otimes} R \underset{\cdot}{\otimes} y$ [62, 65]:

$$(x)^{+}_{R,\varphi} = \bigvee \{ y \in Y \mid \langle x \mid R \mid y \rangle \leq \varphi \} = (x^{\mathsf{t}} \underset{\cdot}{\otimes} R) \underset{\cdot}{\backslash} \varphi = R^{\circledast} \dot{\otimes} x^{-1} \dot{\otimes} \varphi \tag{3.3.3a}$$

$$\overset{+}{}_{R,\varphi}(y) = \bigvee \{ x \in X \mid \langle x \mid R \mid y \rangle \leq \varphi \} = \varphi \underset{\cdot}{/} (R \underset{\cdot}{\otimes} y) = \varphi \dot{\otimes} y^{\circledast} \dot{\otimes} R^{\circledast} \tag{3.3.3b}$$

Algebraicamente[1], dado el vector unitario de un objeto $g_i = [\bot \cdots e \cdots \bot]^{\mathrm{t}} \in \mathbb{R}^{1\times m}$, su intensión es obtenida de (3.3.3a)

$$(g_i)^{+}_{R,\varphi} = R^{\mathrm{t}} \dot{\otimes} g_i^{-1} \dot{\otimes} \varphi = \begin{bmatrix} -r_{11} & \cdots & -r_{i1} & \cdots & -r_{g1} \\ \vdots & & \vdots & & \vdots \\ -r_{1j} & \cdots & -r_{ij} & \cdots & -r_{gj} \\ \vdots & & \vdots & & \vdots \\ -r_{1m} & \cdots & -r_{im} & \cdots & -r_{gm} \end{bmatrix} \dot{\otimes} \begin{bmatrix} \top \\ \vdots \\ e \\ \vdots \\ \top \end{bmatrix} \dot{\otimes} \varphi =$$

$$= \begin{bmatrix} \sum^{\bullet}\{\top - r_{11}, .., e_i - r_{i1}, .., \top - r_{g1}\} \\ \vdots \\ \sum^{\bullet}\{\top - r_{1j}, .., e_i - r_{ij}, .., \top - r_{gj}\} \\ \vdots \\ \sum^{\bullet}\{\top - r_{1m}, .., e_i - r_{im}, .., \top - r_{gm}\} \end{bmatrix} \dot{\otimes} \varphi = \begin{bmatrix} \varphi - r_{i1} \\ \vdots \\ \varphi - r_{ij} \\ \vdots \\ \varphi - r_{im} \end{bmatrix} \tag{3.3.4}$$

Y, usando (3.3.3b) para obtener su extensión, tenemos que:

$$\overset{+}{}_{R,\varphi}((g_i)^{+}_{R,\varphi}) = \varphi \dot{\otimes} \left((g_i)^{+}_{R,\varphi}\right)^{\circledast} \dot{\otimes} R^{\mathrm{t}} \quad = \varphi \dot{\otimes} \left[r_{i1} - \varphi \cdots r_{ij} - \varphi \cdots r_{im} - \varphi\right] \dot{\otimes} R^{\mathrm{t}} \tag{3.3.5}$$

Este proceso puede ser repetido para todos los posibles valores de $i \in \{1 \ldots g\}$ lo que producirá el conjunto de los conceptos-objeto $\{\widetilde{\gamma}^{\varphi}(\mathbf{g}_i)\}^{g}_{i=1} = \left\{\left(\overset{+}{}_{\varphi}((\mathbf{g}_i)^{+}_{\varphi}), \overset{+}{}_{\varphi}(\mathbf{g}_i)\right)\right\}^{g}_{i=1}$ como en (1.4.8).

Análogamente, para el atributo de vector unitario $m_j = \left[\bot \cdots e_j \cdots \bot\right]^{\mathrm{t}} \in \mathbb{R}^{m\times 1}$ la extensión se calcula con (3.3.3b):

$$\overset{+}{}_{R,\varphi}(m_j) = \varphi \dot{\otimes} m_j^{\circledast} \dot{\otimes} R^{\mathrm{t}} = \varphi \dot{\otimes} \begin{bmatrix} \top & \cdots & e_j & \cdots & \top \end{bmatrix} \dot{\otimes} \begin{bmatrix} -r_{11} & \cdots & -r_{j1} & \cdots & -r_{g1} \\ \vdots & & \vdots & & \vdots \\ -r_{1k} & \cdots & -r_{jk} & \cdots & -r_{gk} \\ \vdots & & \vdots & & \vdots \\ -r_{1m} & \cdots & -r_{jm} & \cdots & -r_{gm} \end{bmatrix}$$

$$= \begin{bmatrix} \varphi - r_{1j} \cdots \varphi - r_{ij} \cdots \varphi - r_{gj} \end{bmatrix} \tag{3.3.6}$$

Como en el caso anterior, podemos aplicar el siguiente polar a obtener la intensión para el atributo j y obtener el conjunto de conceptos-atributo $\{\widetilde{\mu}^{\varphi}(\mathbf{m}_j)\}^{m}_{j=1} = \left\{\left(\overset{+}{}_{\varphi}(\mathbf{m}_j), \overset{+}{}_{\varphi}\left((\mathbf{m}_j)^{+}_{\varphi}\right)\right)\right\}^{m}_{j=1}$.

Acorde con la teoría de $\mathcal{K}$-FCA de la Sección 1.4 el φ-retículo está definido como:

$$I^{\varphi}_{R}(i,j) = \widetilde{\gamma}^{\varphi}(\mathbf{g}_i) \leq \widetilde{\mu}^{\varphi}(\mathbf{m}_j), \forall i \in 1 \ldots g, j \in 1 \ldots m \tag{3.3.7}$$

pero es posible sustituir (3.3.5) y (3.3.6) en (3.3.7) para obtener la siguiente condición, por ejemplo, sobre las extensiones:

$$\begin{aligned} &\sum^{\bullet}\{r_{i1} - r_{x1}, \cdots, r_{ij} - r_{xj}, \cdots, r_{ig} - r_{xg}\} \leq \varphi - r_{xj} \\ &\sum^{\bullet}\{r_{i1} - r_{x1} + r_{xj}, \cdots, r_{ij}, \cdots, r_{ig} - r_{xg} + r_{xj}\} \leq \varphi \end{aligned} \tag{3.3.8}$$

[1] A veces, en una expresión usamos mezcla de la notación de semianillos y del álgebra de reales estándar: incluso en este caso se ha elegido la notación de semianillos para que las expresiones no generen ambigüedad.

Esto implica que si $r_{ij} \leq \varphi$ entonces la ecuación 3.3.8 se cumple y $I_R^{\varphi}(i,j) = 1$.

Si existiera un valor r_{ij} tal que $r_{ij} > \varphi$ y $I_R^{\varphi}(i,j) = 1$ implicaría que (3.3.8) es cierta. Esto debería cumplirse para todos los valores de x y n:

$$r_{ij} > \varphi \geq r_{in} - r_{xn} + r_{xj} \qquad \forall n \in 1 \cdots m, x \in 1 \cdots g \tag{3.3.9}$$

Pero (3.3.9) sólo se satisface si $n = j$, así la siguiente relación ha quedado demostrada:

$$\begin{aligned} r_{ij} \leq \varphi &\Leftrightarrow I_R^{\varphi}(i,j) = 1 \\ r_{ij} \not\leq \varphi &\Leftrightarrow I_R^{\varphi}(i,j) = 0 \end{aligned} \tag{3.3.10}$$

Para el semianillo $\overline{\mathbb{R}}_{\text{máx},+}$ la exploración para diferentes valores de φ se hará en el rango $(-\infty, 0]$, puesto que el nivel $\varphi = -\infty$ permitiría ver qué genes no han sido expresados (un caso límite que nunca se da), y el nivel $\varphi = 0$ significa que se consideran valores de expresión que están próximos a los normales, en cuyo punto sus valores son muy ruidosos. Nótese que **los genes más infra-expresados son los que aparecen antes**, en valores más bajos de φ. Claramente, no tiene sentido explorar valores superiores a $\varphi = 0$.

Por lo que respecta al retículo estructural $\underline{\mathfrak{B}}(G, M, I_R^{\varphi})$, a medida que el valor φ aumenta también lo hace el número de elementos de $I_R^{\varphi}(i,j)$ no nulos, lo que potencialmente hace crecer el número de conceptos. En la práctica, el número de conceptos sigue una ley compleja similar a una sigmoide en el dominio $[-\infty, 0]$, que veremos en figuras posteriores.

3.3.3 Análisis de la sobre-expresión con el semicuerpo min-plus

El semicuerpo min-plus $\overline{\mathbb{R}}_{\text{mín},+} = \langle \overline{\mathbb{R}}, \wedge, \dot{+}, -\infty, 0, \infty \rangle$ se utiliza para obtener el grupo de los genes sobre-expresados. Se considera la matriz R parte del contexto $\overline{\mathbb{R}}_{\text{mín},+}$-*concepto formal* $(G, M, R)_{\overline{\mathbb{R}}_{\text{mín},+}}$ con un producto escalar

$$[x \mid R \mid y] = x^{\mathrm{t}} \dot{\otimes} R \dot{\otimes} y\,. \tag{3.3.11}$$

Los mapas adjuntos duales sobre el *orden dual* ahora define el **mínimo grado de existencia requerido** ϕ para que un par de vectores sea considerado un ϕ-concepto.

$$\begin{aligned} (x)^{+}_{R,\phi} &= \bigwedge \{ y \in Y \mid [x \mid R \mid y] \geq \phi \} = R^{\circledast} \dot{\otimes} x^{-1} \dot{\otimes} \phi \\ {}^{+}_{R,\phi}(y) &= \bigwedge \{ x \in X \mid [x \mid R \mid y] \geq \phi \} = \phi \dot{\otimes} y^{\circledast} \dot{\otimes} R^{\circledast} \end{aligned} \tag{3.3.12}$$

Se aplica el mismo proceso de construir conceptos-objeto y -atributo al semianillo $\overline{\mathbb{R}}_{\text{mín},+}$ con el resultado dual:

$$r_{ij} \geq \phi \Leftrightarrow I_R^{\phi}(i,j) = 1 \tag{3.3.13}$$

Así, para encontrar genes sobre-expresados en (G, M, R) es necesario explorar $\underline{\mathfrak{B}}^{\phi}(G, M, R)_{\overline{\mathbb{R}}_{\text{mín},+}}$ con $\phi \in [0, \infty)$. En este caso la exploración va desde ∞ a 0, **los genes más sobre-expresados aparecerán antes** lo que implica valores más altos de ϕ.

3.4 Técnicas de visualización

En el marco de EDA una parte muy importante a la hora de interpretar los datos es la representación gráfica. Existe una gran variedad de técnicas dentro de EDA que permiten representar los datos de diferentes formas como el diagrama de cajas, matrices de dispersión, media polaca, diagrama de tallo y hojas, etc.[5]. Siguiendo esta línea se ha puesto especial interés en generar un conjunto de técnicas de visualización que permitan extraer de la forma más sencilla la información de una exploración $\mathcal{K}$-FCA.

3.4.1 Exploración del número de conceptos

Una importante característica de los retículos de conceptos producida en la fase de exploración es la evolución del número de conceptos con el umbral de existencia, φ, para la exploración $\overline{\mathbb{R}}_{\text{máx},+}$ y ϕ para la $\overline{\mathbb{R}}_{\text{mín},+}$. Esta evolución proporciona una idea de la complejidad de los diferentes retículos de conceptos.

Veamos: por un lado, para observar los genes sobre-expresados hace falta analizar el rango $\phi \in [0, \infty)$ mediante el dominio *min-plus*, mientras que para observar los grupos de genes infra-expresados es preciso analizar el rango $\varphi \in (-\infty, 0]$ mediante $\overline{\mathbb{R}}_{\text{máx},+}$-FCA.

Por otro lado, cada valor de φ diferente genera, potencialmente, un retículo estructural $\underline{\mathfrak{B}}^{\varphi}(G, M, R)_{\overline{\mathbb{R}}_{\text{máx},+}}$ diferente; y lo propio podemos esperar del $\overline{\mathbb{R}}_{\text{mín},+}$-FCA. Por eso, nos esperamos del orden de $\mathcal{O}(g \times m)$ diferentes retículos para una matriz de orden $g \times m$.

Si cuantificamos láxamente la "complejidad cognitiva" de un retículo conceptual como proporcional al número de conceptos de dicho retículo, entonces es posible definir la "complejidad cognitiva" de la exploración de una particular matriz de expresión como la complejidad de la secuencia de retículos al ir variando los umbrales de exploración. Además, el grafo de $|\underline{\mathfrak{B}}^{\varphi}(G, M, R)_{\overline{\mathbb{R}}_{\text{máx},+}}|$ en función de $\varphi \in [-\infty, 0]$ y el de $|\underline{\mathfrak{B}}^{\phi}(G, M, R)_{\overline{\mathbb{R}}_{\text{mín},+}}|$ en función de $\phi \in [0, \infty]$ se pueden solapar, puesto que sus dominios son adyacentes y en ambos casos el rango es el mismo: $|\underline{\mathfrak{B}}^{\varphi}(G, M, R)_{\overline{\mathbb{R}}_{\text{máx},+}}| \in [1, 2^{min(g,m)}]$, y con ello se puede definir la función que describe la evolución del número de conceptos según un umbral genérico $n_{(G,M,R)}(\xi)$:

$$n(G, M, , R)(\xi) = \begin{cases} |\underline{\mathfrak{B}}^{\xi}(G, M, R)_{\overline{\mathbb{R}}_{\text{máx},+}}| & \xi \in [-\infty, 0] \\ |\underline{\mathfrak{B}}^{\phi}(G, M, R)_{\overline{\mathbb{R}}_{\text{mín},+}}| & \xi \in [0, \infty] \end{cases}$$

En la Figura 3.1 [17] se muestra una típica gráfica del número de conceptos. A continuación detallaremos cómo se usa en la exploración de la matriz de expresión.

Típicamente, el máximo número de conceptos se produce para valores de umbral φ y ϕ cercanos 0. Cuanto más alejado sea el valor del umbral de 0 menos conceptos habrá en los retículos, y son estos conceptos que aparecen para umbrales alejados de 0 los que presentarán una mayor robustez y definen grupos de mayor calidad. De especial interés son los valores del umbral para los que se produce un incremento brusco el número de conceptos, o aquellos puntos en los que el número de conceptos parece estabilizarse. Estos puntos indican cambios importantes en la forma de los retículos y deberían ser los primeros en ser estudiados.

En esta tesis hemos puesto especial énfasis en estudiar cómo cambian los grupos y cómo evolucionan los genes en función del valor de umbral. Por ello en la Sección 3.4.3 se muestra

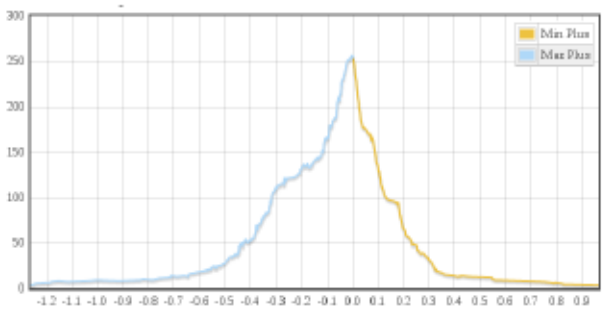


Figura 3.1: Número de conceptos en función del umbral φ (azul claro, a la izquierda de 0,0) y ϕ (amarillo apagado, a la derecha de 0,0) para el contexto explorado. Las posiciones de esta curva pueden ser exploradas de forma iterativa en https://webgenekfca.com/webgenekfca/kfcaresultses/4.

cómo un gen cambia de concepto en función del umbral ϕ seleccionado. El rango de valores de ϕ para los cuales el gen permanece en el mismo grupo indica que ese gen **presenta mayores probabilidades de pertenecer a ese grupo**. También se puede estudiar la evolución de un coagrupamiento en conjunto como se verá en la Sección 3.4.4, donde se ven los genes que aparecen en el grupo seleccionado.

3.4.2 Visualización del retículo

La exploración $\mathcal{K}$-FCA acabará con cientos de retículos conceptuales diferentes, estudiar estos retículos puede ser realmente difícil por lo tanto es necesario encontrar un método que los muestre de una forma que puedan ser fácilmente comparables. Existen diversas herramientas que permiten mostrar un retículo como un diagrama de Hasse siguiendo diferentes reglas basadas en el número de intersecciones de aristas y la atracción/repulsión entre nodos. Una revisión de diferentes técnicas puede encontrarse en [79].

En esta tesis se ha utilizado Conexp[10] en alguna ocasión, como por ejemplo en la figura 1.2. Pero aunque estos algoritmos producen retículos visualmente sencillos de entender los hace bastante malos para ver la evolución de los retículos en función del umbral φ/ϕ porque los conceptos formales cambian de posición de un retículo a otro. Es decir, no se puede saber a priori qué posición ocupará un concepto y por lo tanto es difícil comparar retículos para diferente umbral.

Para resolver este problema decidimos crear un algoritmo que dibuja cada retículo para un umbral dado pero mantiene los conceptos en la misma posición mientras cambia el umbral [64]. Se dibujan en la misma posición los *concepto formales* que para diferente umbral tengan los mismo atributos. Es decir, se trabajará con los *conceptos-atributo* que se definieron en la Sección 1.3.2.

En este punto introducimos la idea del *identificador de concepto* donde cada concepto-atributo en un retículo tiene un *id* diferente dado por sus atributos. El *id* se obtiene asumiendo que un atributo que no está presente tiene un valor 0 y que uno presente tiene un valor 1. Esta secuencia de ceros y unos crea un numero binario que da lugar al *identificador de concepto*. En las figuras 3.2 y 3.3 se muestra dentro de cada concepto su correspondiente identificador.

El algoritmo desarrollado crea un diagrama de Hasse siguiendo dos sencillas reglas:

- El número de 1's en el *identificador de concepto*, el cual es el número de atributos activos, define la posición vertical del concepto.

- Para una misma posición vertical los conceptos son ordenados de izquierda a derecha en orden ascendente dado por su *identificador de concepto*.

Estas dos reglas serán explicadas con más detalle en las siguientes secciones. La figura 3.2 muestra un ejemplo de diagrama de Hasse dibujado siguiendo este algoritmo. El radio de cada concepto es proporcional al número de objetos que pertenecen de forma exclusiva al concepto, una de los criterios habituales para estos diagramas.

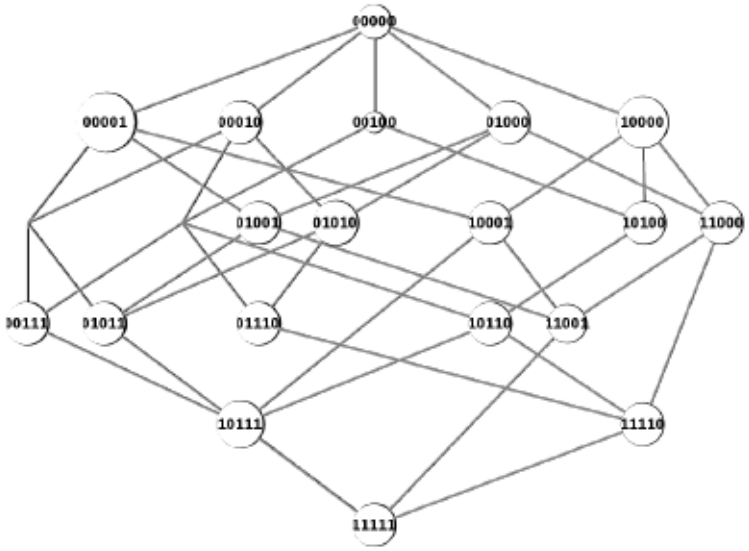


Figura 3.2: Ejemplo de retículo dibujado con el algoritmo descrito en 3.4.2. Los conceptos que aparecen en este retículo están en la misma posición que aquellos mostrados en el retículo completo de la Figura 3.3.

3.4.2.1 Posición vertical

Este método puede ser estudiado como si empezáramos con el diagrama de Hasse de un retículo FCA completo con todos los posibles conceptos formales representados, es decir el retículo booleano de orden $|M| = m$. Por ejemplo, un retículo booleano con cinco atributos se mostraría tal y como aparece en la figura 3.3. Cada nodo tiene una posición fija dada por sus atributos.

Este retículo puede ser dividido en tantos niveles como atributos tiene más uno. En el primer nivel sólo hay un concepto, el concepto *supremo* ($\top$) que no tiene atributos. En el segundo nivel los conceptos tienen un atributo. El tercer nivel tiene los conceptos con dos atributos y así hasta el último nivel donde solo hay un concepto, el ínfimo ($\bot$) que tiene todos los atributos.

El número de conceptos posibles sen cada nivel viene dado por la combinación del número

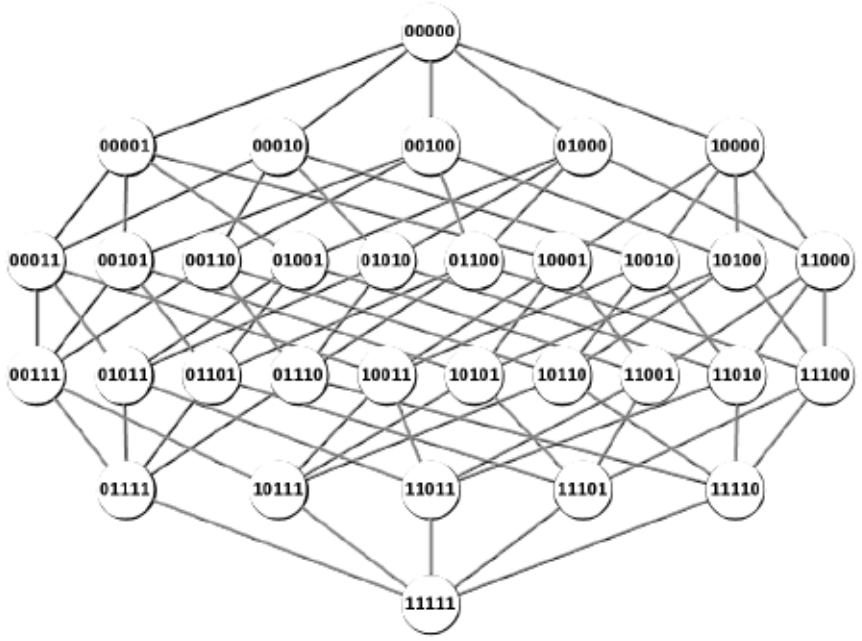


Figura 3.3: Retículo con todos los conceptos posibles. los conceptos en cada nivel están ordenados por su identificador de concepto.

de atributos activos y el número total de atributos sin repetición:

$$nCr(|M|, l) = \binom{|M|}{l} = \frac{|M|!}{l!\,(|M|-l)!} = \prod_{i=1}^{l} \frac{|M|-l+i}{i} \tag{3.4.1}$$

Donde $|M|$ es el número de atributos que el concepto tiene y l es el número de nivel, donde el concepto $\top$ está en el nivel 0. Así para un análisis FCA con cinco atributos el nivel 1 del retículo (el que está justo debajo de $\top$) tiene cinco posibles conceptos, el nivel 2 y 3 tienen diez y el nivel 4 tiene cinco conceptos.

El identificador de concepto aumenta de arriba a abajo y de izquierda a derecha, los hijos de un concepto siempre tienen un identificador mayor que sus padres.

3.4.2.2 Posición horizontal

La posición horizontal de un concepto viene dada por su *identificador de concepto*, abreviado de ahora en adelante como *id*. Para dos conceptos con el mismo número de atributos se coloca a la derecha el que tiene un identificador de concepto más alto. Por cada nivel hay un máximo número de conceptos dado por (3.4.1), así que una vez conocido el identificador de concepto el problema consiste en obtener su posición dentro del intervalo $[0, nCr(|M|, l) - 1]$. Esta posición puede ser obtenida mediante un algoritmo iterativo:

1. Primero se comienza con el *id* más pequeño para el nivel dado. Esto es, un *id* con todos los $1's$ en las posiciones menos significativas. Por ejemplo para un concepto en el nivel 3 se empezaría con un id $111_{(2)} = 7_{(10)}$.

pos	*id*	s_2	s_1	s_0
0	$00111_{(2)}$	0	0	0
1	$01011_{(2)}$	1	1	0
2	$01101_{(2)}$	1	1	0
3	$01110_{(2)}$	1	1	1
4	$10011_{(2)}$	2	0	0
5	$10101_{(2)}$	2	1	0
6	$10110_{(2)}$	2	1	1
7	$11001_{(2)}$	2	2	0

Tabla 3.2: Método para calcular la posición horizontal del concepto con id $25_{(10)} = 11001_{(2)}$. La primera columna (pos) indica la posición del concepto dado por la segunda columna. Las últimas columnas (s_i) muestran cuantas veces los $1's$ han tenido que ser desplazados desde su posición original en la fila 0.

2. Desplazar el 1 menos significativo que tiene un 0 pegando a su izquierda una posición a la izquierda. Por ejemplo si el *id* fuera $0111_{(2)}$ después de este paso se convertiría en $1011_{(2)} = 11_{(10)}$.
3. Repetir el paso 2 hasta obtener el *id* deseado. La posición horizontal del concepto viene dada por el número de veces este paso se ha repetido.

La ejecución de este algoritmo se puede entender mejor con el ejemplo mostrado en la tabla 3.2. Para el identificador de concepto $25_{(10)} = 11001_{(2)}$ su posición vertical sería 3 y su posición horizontal 7.

El problema de este algoritmo es su complejidad: $O(m^3)$ con m el número de atributos, lo que ralentiza mucho el generar el diagrama. Sin embargo, es posible obtener una formula polinómica para calcular la posición horizontal basada en el número de desplazamientos necesitado, $[s_{n-1}, s_{n-2}, \dots, s_0]$, por cada 1 hasta llegar a la posición final.

Así para llegar a $11001_{(2)}$ desde $00111_{(2)}$ el 1 menos significativo (s_0) no necesita ningún desplazamiento, y los otros dos $1's$ en las posiciones s_1 y s_2 necesitan un desplazamiento de 2 posiciones, lo cual genera el array de desplazamientos $[2, 2, 0]$. Este proceso también se muestra en la tabla 3.2 en las columnas $[s_2, s_1, s_0]$.

Entonces la posición de un concepto situado verticalmente en el nivel 3, con tres atributos activos, viene dado por la fórmula:

$$pos(s_2, s_1, s_0) = \sum_{n_1=0}^{s_2} \sum_{i=0}^{n_1} i + \sum_{i=0}^{s_1} i + s_0 \tag{3.4.2}$$

donde s_i es el número de posiciones del i-ésimo 1 es desplazado. En este ejemplo con id $11001_{(2)}$ sería $pos(2, 2, 0) = 4 + 3 + 0 = 7$.

La suma de sumas puede ser transformada en un coeficiente binomial [80] con $\binom{n+k}{n-1}$, donde n es el máximo número a sumar y k es el número de sumatorios. Así la expresión para calcular la posición horizontal se simplifica en:

$$pos(s_{l-1}, s_{l-2}, \dots, s_0) = \sum_{i=0}^{l-1} \binom{s_i + i}{s_i - 1} = \sum_{i=0}^{l-1} nCr(s_i + i, s_i - 1) \tag{3.4.3}$$

donde l es el número de unos en el identificador de concepto y s_i es el número de desplazamientos necesitado por el i-ésimo 1 para alcanzar su posición final.

3.4.2.3 Dibujando el retículo

La parte más difícil cuando se dibuja un retículo consiste en saber dónde dibujar los conceptos, problema que ha sido resuelto en las secciones previas. El siguiente paso consiste en dibujar, primero, todos los conceptos del retículo, incluso aquellos que no tienen objetos en exclusiva, y dibujar, luego, vértices entre ellos para mostrar las relaciones padre-hijo.

Pero en un diagrama de Hasse no todos los conceptos tienen objetos en exclusiva: por ejemplo en el retículo 3.2 el concepto $00011_{(2)}$ es el padre del $00111_{(2)}$ y $01011_{(2)}$, pero no tiene ningún elemento en exclusiva.

Estos conceptos se tienen que encontrar y tienen que ser dibujados como vértices en el diagrama de Hasse. Estos elementos vacíos pueden ser encontrados buscando por padres comunes entre dos conceptos diferentes con el operador $AND = \wedge$.

Una vez que todos los conceptos han sido identificados se pintan en su correspondiente posición dada por su identificador. El siguiente paso consiste en pintar las aristas entre dos conceptos que tienen una relación padre-hijo. Un concepto con *id* a es ascendiente de un concepto con *id* b si y solo si $a \wedge b = a$. Entonces una arista es dibujada entre un concepto y todos sus ascendientes. Este proceso crea un retículo con muchas aristas redundantes, algunas de las cuales tienen que ser eliminadas. Si a es padre de b y b es padre de c, entonces la línea entre a y c tiene que ser eliminada. Esta poda de aristas es representada en la Figura 3.4 donde las lineas discontinuas son las aristas eliminadas. Una vez acabado este paso el diagrama de Hasse está concluido.

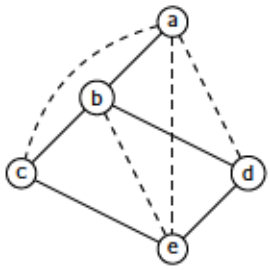


Figura 3.4: Construcción del diagrama de Hasse. Cada circulo es un concepto y las lineas muestran la relación padre-hijo. Las lineas discontinuas son redundantes y tienen que ser eliminadas.

3.4.3 Evolución de objetos en el retículo

Esta forma de dibujar un retículo, con las intensiones en la misma posición, tiene diferentes ventajas, no solo para comparar diferentes retículos en función del umbral seleccionado, ya que los nodos que representan los conceptos siempre mantienen una posición fija, sino también para ver cómo cambia un objeto de un concepto a otro a medida que cambia el umbral. En el

caso del análisis de matrices de expresión de genes, este proceso muestra el cambio de grupo para un gen a medida que se cambia el umbral.

Para conseguir este efecto se comienza a crear el retículo FCA dibujando el concepto supremo ($\top$) junto con todos los nodos del nivel 1 que podrían existir para este retículo. A continuación se van dibujando nodos, uno en cada uno de los siguientes niveles, correspondientes a los conceptos en los que el objeto en cuestión podría estar. Se puede indicar el valor del umbral a partir del cual el objeto entra en ese concepto. Además el radio del nodo es proporcional al rango en el cual el objeto está en el concepto correspondiente de forma exclusiva. Por último se dibuja el concepto ínfimo ($\bot$) aunque en la exploración el objeto nunca haya llegado a formar parte de ese concepto.

Por ejemplo supongamos un objeto A bajo exploración min-plus que para $\phi = 0$ está en el concepto ínfimo y para $\phi = 7$ está en el supremo. Su viaje a lo largo de los conceptos se podría resumir en la siguiente tabla:

id	ϕ
$\top$	$[7, \infty)$
$00010_{(2)}$	$[6, 7)$
$00110_{(2)}$	$[4, 6)$
$00111_{(2)}$	$[3, 4)$
$10111_{(2)}$	$[1, 3)$
$\bot$	$[0, 1)$

El dibujo de este concepto moviéndose a lo largo del retículo FCA se muestra en la Figura 3.5.

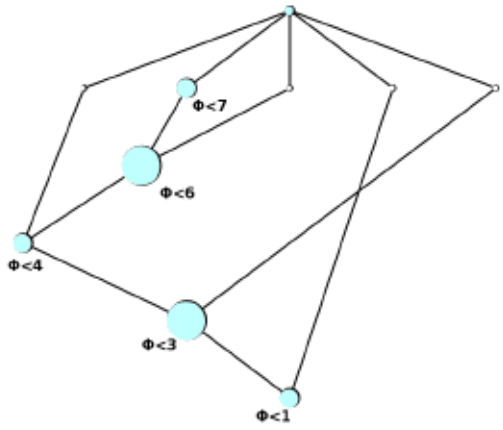


Figura 3.5: Variación de un objeto a lo largo del retículo.

Esto sirve para ver qué relación tiene el objeto seleccionado con cada concepto. Por ejemplo no es lo mismo que el objeto adquiera los atributos del concepto en el orden $a1 \rightarrow a2 \rightarrow a3$ que $a1 \rightarrow a3 \rightarrow a2$. En el primer caso significaría que, posiblemente, esté más relacionado con la condición a2 que con la a3.

3.4.4 Evolución del grupo en la exploración

A veces es útil simplemente centrarse en un coagrupamiento de interês. Por ejemplo, si se sabe a priori que ciertas muestras de la matriz GED pertenecen a un grupo de interés (ej: células cancerosas) que se desea estudiar con detalle, es posible fijarse únicamente en los genes que entran en el concepto formal dado por estas características y así identificar los genes que únicamente se ven afectados en ese tejido de estudio.

Imaginemos la siguiente matriz de expresión genética, las filas muestran los genes etiquetados $g1 - g5$ y las columnas representan las muestras analizadas, etiquetadas como $c1 - c4$ y supongamos que durante la exploración $\overline{\mathbb{R}}_{\text{mín},+}$, queremos centrarnos únicamente en el grupo dado por las condiciones $c1$ y $c2$.

	$c1$	$c2$	$c3$	$c4$
$g1$	7	5	-6	-6
$g2$	7	4	-12	1
$g3$	5	3	-8	0
$g4$	4	3	-2	-5
$g5$	-2	-1	0	3

Se ve que el gen $g1$ en el rango $[0, 5]$ tendrá activos únicamente los atributos $c1$ y $c2$. Los genes $g3$ y $g4$ tendrán los atributos $c1$ y $c2$ activos durante un rango menor, $[0, 3]$, lo que indica que $g1$ tiene una mayor dependencia con las condiciones $c1$ y $c2$ que cualquiera de los otros genes. En cambio el gen $g2$ pertenecerá en exclusiva al concepto generado por $c1$ y $c2$ en el rango $[1, 4]$, en el rango $[0, 1]$ el atributo $c3$ aparece asociado a este gen en el retículo de conceptos. Esto indica que aunque se ve influenciado por $c1$ y $c2$ también parece influenciado aunque en menor medida por $c3$. Dependiendo del experimento que se esté realizando y de cuales sean las implicaciones biológica de $c3$ esto tendrá más o menos importancia.

Para facilitar este tipo de análisis se ha propuesto un nuevo tipo de diagrama que se muestra en la Figura 3.6. En la figura 5.11 aparece la misma visualización para un ejemplo real. Como verá en el Capítulo 5 cuanto mayor sea el rango de valores de ϕ/φ para los cuales un gen permanezca en exclusiva en un concepto formal, más probabilidades habrá de que dicho gen esté correctamente clasificado en ese grupo.

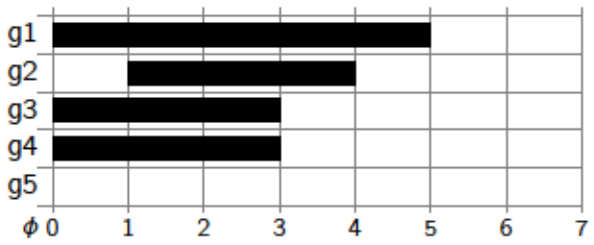


Figura 3.6: Evolución de los genes dentro de un grupo en función de ϕ

3.5 Ejemplo de interpretación con KFCA de GED idealizados

Hasta ahora se ha explicado la teoría detrás de $\mathcal{K}$-FCA y el papel de los umbrales φ y ϕ para analizar infra- y sobre-expresión. En esta sección se tratará de dar ese significado práctico, haciendo hincapié en el procedimiento EDA que proponemos, partiendo de una matriz de ejemplo de datos idealizados. En los Capítulos 4 y 5 presentaremos los análisis sobre matrices in silico y de datos reales, respectivamente.

Supongamos que tras analizar diferentes muestras de tejidos bajo diferentes condiciones obtenemos los niveles de expresión de 10 genes. Esta hipotética matriz de expresión genética se muestra en la Tabla 3.3 en donde cada elemento de la matriz se corresponde con el logaritmo del nivel de concentración de las cadenas de ARNm correspondientes a cada gen.

	c1	c2	c3	c4	c5	c6	c7	c8
genA	1	1	1	1	-0.5	-0.5	-0.5	-0.5
genB	1	1	1	1	-0.5	-0.5	-0.5	-0.5
genC	2	2	2	2	-1	-1	-1	-1
genD	2	2	2	2	-1	-1	-1	-1
genE	-0.5	-0.5	-0.5	-0.5	1	1	1	1
genF	-0.5	-0.5	-0.5	-0.5	1	1	1	1
genG	-2	-2	-2	-2	1	1	1	1
genH	-2	-2	-2	-2	1	1	1	1
genI	-0.5	-0.5	-1	-1	-1	-1	-0.5	-0.5
genJ	-0.5	-0.5	1	1	1	1	-0.5	-0.5

Tabla 3.3: Ejemplo de expresión genética para diez genes en el rango genA-J y ocho condiciones diferentes c1-c8. Los valores son el logaritmo de concentraciones de cadenas ARNm.

Aunque el significado de φ/ϕ depende en gran medida de la normalización realizada primero trataremos de hacer una generalización ignorando la normalización.

La exploración comienza en el dominio *max-plus* para revelar los genes infra-expresados. Para este caso de análisis, según aumenta el valor de φ los objetos ocupan conceptos situados cada vez más abajo, porque los objetos irán ganando más atributos. Este barrido va desde que todos los objetos no tienen ningún atributo asociado (están en el concepto supremo $\top$) hasta que, poco a poco, más y más objetos van entrando en el concepto ínfimo ($\bot$), es decir, tienen todos los atributos. Algunos de los retículos que se obtienen durante esta exploración se pueden ver en la Figura 3.7.

La exploración comienza con $\varphi = -2$ de forma que la matriz de Tabla 3.3 se convierte en una matriz booleana (Tabla 3.4) donde las celdas con valor menor o igual a -2 consideran que el gen contiene el atributo marcado por la columna. En este retículo hay tres conceptos formales tal y como se muestra en la Tabla 3.5: el supremo $\top$, que agrupa todos los genes, de los cuales 8 genes (A, B, C, D, E, F, I, J) en exclusiva sólo pertenecen a este concepto; otro, que agrupa los genes G y H que aparecen fuertemente infra-expresados para las condiciones 1-4, y el concepto formal ínfimo $\bot$, que no posee ningún gen.

A medida que aumenta el valor de φ típicamente aumenta el número de conceptos: en $\varphi = -1$ hay 7 conceptos. En la Tabla 3.6 se muestra cómo es la matriz booleana para este

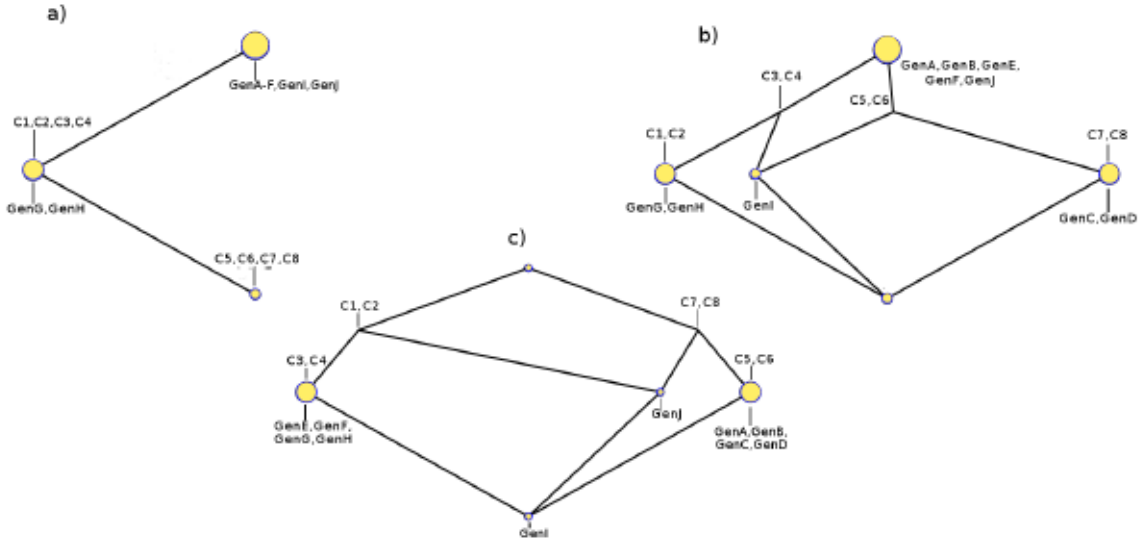


Figura 3.7: Diagramas de diferentes retículos para a) $\varphi = -2$, b) $\varphi = -1$, c) $\varphi = 0$

$\mathbb{R}_{máx,+}$	c1	c2	c3	c4	c5	c6	c7	c8
genA								
genB								
genC								
genD								
genE								
genF								
genG	x	x	x	x				
genH	x	x	x	x				
genI								
genJ								

Tabla 3.4: Matriz booleana $\varphi = -2$.

Conceptos formales
$(\{genA, genB, genC, genD, genE, genF, genG, genH, genI, genJ\}, \emptyset)$
$(\{genG, genH\}, \{c1, c2, c3, c4\})$
$(\emptyset, \{c1, c2, c3, c4, c5, c6, c7\})$

Tabla 3.5: Conceptos formales $\varphi = -2$.

$\mathbb{R}_{máx,+}$	c1	c2	c3	c4	c5	c6	c7	c8
genA								
genB								
genC					x	x	x	x
genD					x	x	x	x
genE	x	x	x	x				
genF	x	x	x	x				
genG	x	x	x	x				
genH	x	x	x	x				
genI	x	x	x	x	x	x	x	x
genJ	x	x					x	x

Tabla 3.6: Matriz booleana $\varphi = -1$.

Conceptos formales
$(\{genA, genB, genC, genD, genE, genF, genG, genH, genI, genJ\}, \emptyset)$
$(\{genG, genH, genI\}, \{c3\})$
$(\{genC, genD, genI\}, \{c5\})$
$(\{genG, genH\}, \{c1, c2, c3, c4\})$
$(\{genI\}, \{c3, c4, c5, c6\})$
$(\{genC, genD\}, \{c5, c6, c7, c8\})$
$(\emptyset, \{c1, c2, c3, c4, c5, c6, c7\})$

Tabla 3.7: Conceptos formales $\varphi = -1$.

$\mathbb{R}_{máx,+}$	c1	c2	c3	c4	c5	c6	c7	c8
genA					x	x	x	x
genB					x	x	x	x
genC					x	x	x	x
genD					x	x	x	x
genE	x	x	x	x				
genF	x	x	x	x				
genG	x	x	x	x				
genH	x	x	x	x				
genI	x	x	x	x	x	x	x	x
genJ	x	x					x	x

Tabla 3.8: Matriz booleana, $\varphi = 0$.

Conceptos formales
$(\{genA, genB, genC, genD, genE, genF, genG, genH, genI, genJ\}, \emptyset)$
$(\{genE, genF, genG, genH, genI, genJ\}, \{c1, c2\})$
$(\{genA, genB, genC, genD, genJ\}, \{c7, c8\})$
$(\{genE, genF, genG, genH, genI\}, \{c1, c2, c3, c4\})$
$(\{genJ\}, \{c1, c2, c7, c8\})$
$(\{genA, genB, genC, genD\}, \{c5, c6, c7, c8\})$
$(\{genI\}, \{c1, c2, c3, c4, c5, c6, c7\})$

Tabla 3.9: Conceptos formales, $\varphi = 0$.

umbral, que se corresponde con el retículo de la Figura 3.7b, y en la Tabla 3.7 se muestran los correspondientes conceptos formales. En este caso el grupo para las condiciones C1-4 mantiene los genes G y H. El supremo pierde varios genes que pasan a formar parte de otros grupos, como los genes C y D que aparecen infra-expresados bajo las condiciones C5-8.

Al continuar aumentando el valor de φ se llega a $\varphi = 0$ donde aparecen 7 conceptos formales, en la Figura 3.7c. Nótese que este no es un caso típico para datos reales, mucho más ruidosos. En la Tabla 3.8 se muestra la matriz booleana correspondiente y en la Tabla 3.9 los conceptos correspondientes. El grupo de las condiciones C1-4 gana un par de elementos quedándose con los genes E, F, G, H e I; lo mismo le ocurre al grupo de condiciones C5-8 que para este umbral también incrementa su tamaño y contiene A, B, C, D e I.

El mismo proceso se repite para analizar la sobre-expresión mediante el dominio min-plus. En este caso la exploración comienza en el valor máximo de la matriz y acaba en 0. Para la sobre-expresión, según disminuye el valor del umbral y más cerca de 0 nos encontramos, los genes se desplazan del supremo al concepto ínfimo. En la figura 3.8 se muestran los retículos para $\phi = 2$ y $\phi = 0$.

Para $\phi = 2$ el número de conceptos formales es bastante bajo, en la Tabla 3.10 se muestra la matriz booleana correspondiente, con sólo 3 conceptos (Tabla 3.11), de los cuales uno es el $\top$ y otro el $\bot$. El único concepto de interês es el de los genes C y D en las condiciones C1-4. De esta forma se puede ver que estos dos genes están fuertemente expresados en esas condiciones.

En el dominio *max-plus* para $\varphi = 0$ los genes infra-expresados únicamente en las condiciones C1-4 son los genes E, F, G y H que a su vez están sobre-expresados para las condiciones C5-8 como se muestra en el retículo para $\phi = 0$ (Figura 3.8a). En el gen J se observa el mismo comportamiento, el hecho de estar infra-expresado para las condiciones C3, C4, C5 y

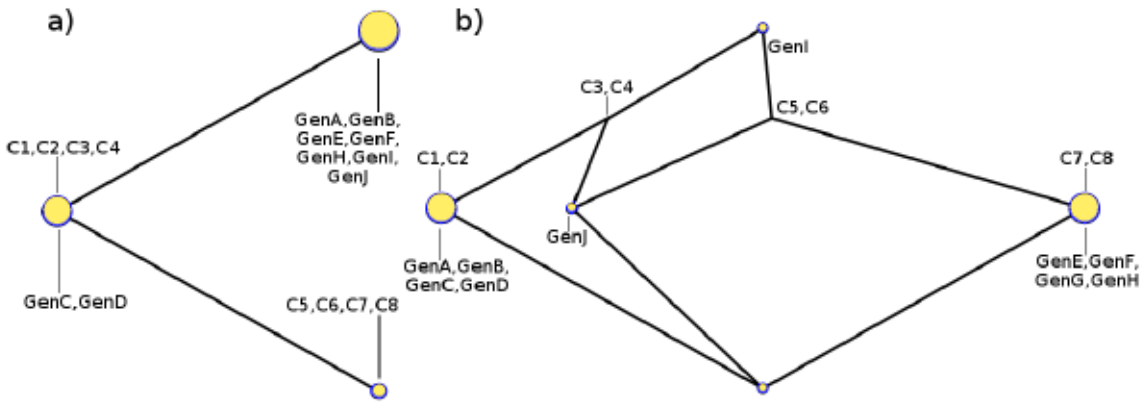


Figura 3.8: Diagramas de diferentes retículos para $\phi = 2$(izquierda). $\phi = 0$(derecha)

$\mathbb{R}_{min,+}$	c1	c2	c3	c4	c5	c6	c7	c8
genA								
genB								
genC	x	x	x	x				
genD	x	x	x	x				
genE								
genF								
genG								
genH								
genI								
genJ								

Tabla 3.10: Matriz booleana, $\phi = 2$.

Conceptos formales
$(\{genA, genB, genC, genD, genE, genF, genG, genH, genI, genJ\}, \emptyset)$
$(\{genC, genD\}, \{c1, c2, c3, c4\})$
$(\emptyset, \{c1, c2, c3, c4, c5, c6, c7\})$

Tabla 3.11: Conceptos formales, $\phi = 2$.

C6 significa que estará sobre-expresado para las condiciones duales C1, C2, C7 y C8. Esto se puede comprobar fácilmente comparando la matriz 3.12 que es la dual de 3.8.

De esta forma el análisis *max-plus* y el *min-plus* en $\varphi = 0 = \phi$ se puede ver como complementario el uno del otro presentando información similar de forma diferente. Por supuesto, la evolución de los grupos de genes en uno y otro polo no están, en general relacionadas, y tienen informaciones diferentes.

Se puede considerar un concepto FCA como un co-grupo (ing. "co-cluster"), ya que los objetos (genes) tienen los atributos que se desean, pero además de tener esos atributos también podrían tener otros extra que hay que tener en cuenta a la hora de realizar el análisis. Por eso en algunas ocasiones será deseable considerar utilizar únicamente los objetos (genes)

$\mathbb{R}_{min,+}$	c1	c2	c3	c4	c5	c6	c7	c8
genA	x	x	x	x				
genB	x	x	x	x				
genC	x	x	x	x				
genD	x	x	x	x				
genE					x	x	x	x
genF					x	x	x	x
genG					x	x	x	x
genH					x	x	x	x
genI								
genJ			x	x	x	x		

Tabla 3.12: Matriz booleana $\phi = 0$.

Conceptos formales
$(\{genA, genB, genC, genD, genE, genF, genG, genH, genI, genJ\}, \emptyset)$
$(\{genA, genB, genC, genD, genJ\}, \{c3\})$
$(\{genE, genF, genG, genH, genI, genJ\}, \{c5, c6\})$
$(\{genA, genB, genC, genD\}, \{c1, c2, c3, c4\})$
$(\{genJ\}, \{c3, c4, c5, c6\})$
$(\{genE, genF, genG, genH\}, \{c5, c6, c7, c8\})$
$(\emptyset, \{c1, c2, c3, c4, c5, c6, c7\})$

Tabla 3.13: Conceptos formales $\phi = 0$.

que contienen únicamente los atributos buscados. Esto se explica con más detalle y ejemplos en el capítulo 4.

3.6 Medidas de calidad de un grupo

Una vez que se tiene identificado un coagrupamiento de genes y muestras puede ser deseable saber cómo de fiable es la hipótesis de que esos genes en esas condiciones estén relacionados. Por ello se dispone de diferentes técnicas que permiten inferir la calidad de un grupo y ayudar en el proceso EDA a seguir profundizando en una dirección u otra.

3.6.1 Homogeneidad y separación

Una de las primeras formas que se nos pueden ocurrir para realizar este tipo de medidas es usar métricas ya utilizadas en otros campos para medir la calidad de los grupos como podría ser la homogeneidad y la separación [73, 81, 82].

La **homogeneidad** mide la distancia intra-grupo, es decir, cómo de juntos están entre sí los elementos de un grupo. La fórmula que hemos utilizado es [73]:

$$H(C) = \frac{1}{|C|} \sum_{g_i, g_j \in C} f_{dist}(g_i, g_j) \tag{3.6.1}$$

donde C es el grupo del cual se quiere conocer su valor de homogeneidad, $|C|$ es el número de elementos en el grupo (genes) y f_{dist} es la medida de separación que tiene el grupo. Cuanto mayor sea su valor, en teoría, mejor debería ser la calidad del grupo.

La **separación** mide la distancia inter-grupos, es decir, cómo de separados están los grupos, C y C_k, entre sí:

$$\begin{aligned} S(C, C_k) &= \frac{\sum_{g_i \in C, g_j \in C_k} f_{dist}(g_i, g_j)}{|C| \cdot |C_k|} \\ S(C) &= \frac{1}{|\mathcal{C}|} \sum_{k \in \mathcal{C}} S(C, C_k) \end{aligned} \tag{3.6.2}$$

Donde $\mathcal{C}$ es el conjunto de grupos que se han encontrado y $|\mathcal{C}|$ representa el cardinal de este conjunto.

Se han utilizado dos medidas de distancia una basada en la **distancia euclídea** (3.6.3a) y otra basada en la **correlación** (3.6.3b):

$$f_{dist}(g_i, g_j) = \sqrt{\sum_{k \in \mu(g_i)} (g_{ik} - g_{jk})^2} \tag{3.6.3a}$$

$$f_{dist}(g_i, g_j) = \frac{cov(g_{ik}, g_{jk})}{\sigma_{g_{ik}} \sigma_{g_{ik}}} \qquad k \in \mu(g_i) \tag{3.6.3b}$$

Estas medidas de distancia sólo consideran las componentes de los perfiles de expresión de los genes g_x en los atributos presentes en el co-agrupamiento. Se ignoran las otras coordenadas fuera del co-agrupamiento que pueden incluir ruido.

En teoría cuanto menor sea el valor de homogeneidad más cerca estarán los elementos de un grupo entre sí, por lo tanto mejor será la calidad del grupo. Cuanto mayor sea el valor de separación más diferentes serán los elementos de un grupo respecto a los de otros.

3.6.2 Enriquecimiento de datos

Otra aproximación para evaluar la calidad de los co-agrupamientos utilizar información biológica conocida a priori. Estos tests se realizan mediante la distribución hipergeométrica como ya se adelantó en la Sección 2.2.

Dado un grupo con n genes se desea saber la probabilidad de que k de ellos pertenezcan a la misma categoría funcional de GO. Para empezar supongamos que tenemos la matriz de clasificación como la que se muestra en la Tabla 3.14. En esta tabla se ve cómo los k genes que comentábamos pertenecen a la categoría funcional deseada y a genes del grupo no pertenecen a dicha categoría. También sabemos que en total hay f genes en todo el genoma que pertenecen a la categoría funcional deseada y $a+c$ genes que no pertenecen. En total el genoma analizado tiene g genes.

	$\in$ **GOTerm**	$\notin$ **GOTerm**	**Total**
grupo	k	$n-k=a$	$n=a+k$
resto	$f-k=b$	c	$g-n=b+c$
Total	$f=k+b$	$g-f=a+c$	$g=n+b+c=f+a+c$

Tabla 3.14: Clasificación de genes en función de la categoría funcional GO.

Asumimos la hipótesis mula H_0 que supone que la probabilidad de que un gen en el grupo pertenezca a la categoría funcional descrita no depende del grupo en el que esté. La probabilidad de que esta hipótesis se cumpla viene dada por el test de Fisher, calculado mediante la distribución hipergeométrica:

$$P(H_0 = k) = \frac{\binom{n}{k}\binom{g-n}{f-k}}{\binom{g}{f}} = \frac{\binom{f}{k}\binom{g-f}{n-k}}{\binom{g}{n}} \tag{3.6.4}$$

La suma de todas las probabilidades para valores de k desde 0 hasta el número de genes en el grupo o el número genes en la categoría funcional, tiene que dar 1 porque cubre todo el abanico de posibilidades:

$$\sum_{k=0}^{\min\{n,g\}} \frac{\binom{f}{k}\binom{g-f}{n-k}}{\binom{g}{n}} = 1 \tag{3.6.5}$$

A partir de la fórmula (3.6.4) en [83] se proponen varios tests estadísticos unilaterales o bilaterales.

En [47, 48, 73] se propone utilizar un test estadístico unilateral definido como la probabilidad de encontrar al menos k genes de una categoría funcional, asumiendo como hipótesis nula que el grupo no clasifica por categoría funcional:

$$P(K \geq k) = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} \tag{3.6.6}$$

En [48] van un paso más allá y además de proponer este test utilizan una simulación donde generan 1000 grupos aleatorios y observan cuál es el porcentaje de grupos que ofrecen una probabilidad inferior a la dada por (3.6.6).

3.7 Conexión con sistemas externos

En esta Sección analizamos una potencialidad de las descritas en la Sección 1.5 que no se ha obtenido directamente de la metáfora de paisajes de conocimiento de Wille. Viene inspirado por el trabajo de Godin [84] quien sugirió que un concepto puede ser convertido en un índice para un sistema de recuperación de información.

En esta tesis, como parte del desarrollo de la herramienta *webgeneKFCA* (ver Apéndice C) hemos incluido la conexión con diferentes bases de datos externas:

- para el caso de *microarrays* se han incorporado las bases de datos exportadas por `Affymetrix` en sus ficheros de anotación[2].
- Para el caso de RNASeq se han utilizado los ficheros de anotaciones de GO[3]. Esta información se guarda en una base de datos local y contiene el nombre de diferentes genes y una breve descripción de ellos junto con identificadores de a qué ontología genética[4](GO) [49] pertenecen y si forman parte de alguna vía metabólica conocida por KEGG[5] [85].

Así pues en la vista web se ponen enlaces a National Center for Biotechnology Information (NCBI) que permite tener información más detallada de un gen en cuestión. También se utiliza la información de ontologías para medir la calidad de los grupos al suponer que los genes agrupados bajo un mismo coagrupamiento comparten términos GO, esto se desarrolla con más detalle en la Sección 3.6.

Además se pueden establecer relaciones entre los diferentes términos GO. Cada gen tiene asociado un conjunto de términos GO y los genes muchas veces comparten términos GO cuando son similares. Muchas veces estos términos guardan entre si una relación jerárquica, lo que se puede aprovechar para tener una visión más general de la relación entre genes y términos GO. Con esta información se puede construir una matriz booleana donde las columnas sean los términos GO y las filas los genes. Cuando un gen está asociado con un termino GO aparecerá un 1 en el elemento de la matriz correspondiente. Esta matriz booleana se puede representar como un retículo FCA lo cual a su vez nos proporcionará más información acerca de los genes, los términos GO y la relación que existe entre ellos como se muestra en la Figura 5.3.

Esta relación entre FCA, $\mathcal{K}$-FCA y las fuentes externas se puede puede interpretar como el diagrama 3.9. Donde un concepto formal se puede usar como índice para obtener información de la base de datos local la cual a su vez con ayuda de sistemas externos como GO completa el significado del concepto proporcionando el conocimiento que se obtiene de tratar toda esa información.

[2] http://www.affymetrix.com/support/technical/annotationfilesmain.affx
[3] http://geneontology.org/page/download-annotations
[4] http://geneontology.org/
[5] http://www.genome.jp/kegg/

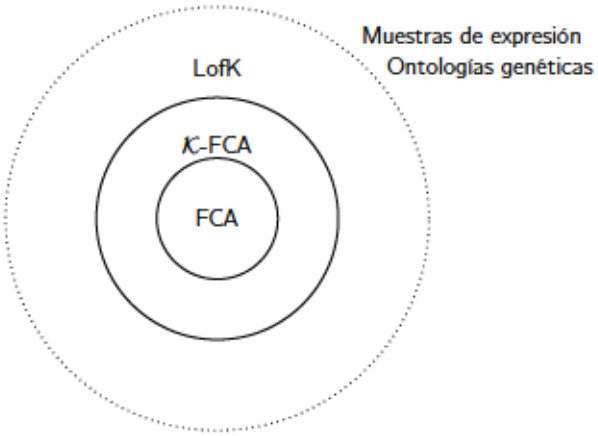


Figura 3.9: Interpretación de la relación entre FCA, $\mathcal{K}$-FCA, "Landscapes of Knowledge" (LofK) y fuentes externas. En el ejemplo las ontologías genéticas estarían indexadas por FCA, pero el acceso a los datos experimentales de expresión usaría una base de datos relacional clásica, o un repositorio web.

3.8 Discusión: factibilidad del análisis exploratorio de GED con $\mathcal{K}$-FCA: y una propuesta de metodología

La relación entre los genes y las muestras dada por la GED puede verse como una conexión de Galois cuyos polares permiten pasar de un dominio a otro y generar pares de conjuntos definidos por muestras y genes denominados conceptos formales. Los conceptos formales se pueden considerar unidades de conocimiento que se relacionan entre sí formando un retículo que puede ser interpretado como mapa a través del cual se puede explorar para extraer la información deseada. Este método de descubrimiento de información sin hacer ninguna suposición sobre los datos de entrada (como distribución esperada, relación entre muestras o genes) encaja perfectamente con la definición dada por Tukey para el EDA.

Un problema del FCA estándar es que no proporciona ninguna protección contra el ruido, es decir que parte de matrices booleanas sin ruido. Por ello en esta tesis hemos decidido utilizar $\mathcal{K}$-FCA como extensión a FCA de forma que se pueda trabajar con datos de GED reales. $\mathcal{K}$-FCA nos proporciona la base para llevar a cabo una exploración de la matriz GED ya que incluye umbrales que proporcionan cierta robustez frente al ruido.

Se podría considerar la extracción de este conocimiento como la exploración de un terreno desconocido con ayuda de un mapa del terreno a diferentes escalas que serían los diferentes retículos de conceptos obtenidos con diferentes umbrales.

Esto permite crear un marco genérico para lidiar con datos en dos dominios (genes y muestras) al que proporciona un contexto, lo que se aprovechará para agrupar subconjuntos de genes en función de su nivel de expresión para un subconjunto de muestras.

Aquí aparecen dos formas de agrupar los genes y muestras como se ha explicado en la Sección 3.5: una sería simplemente mediante el uso de conceptos formales donde sólo se mira si un gen se encuentra sobre/infra-expresado en unas muestras en cuestión sin importar lo que indiquen las otras muestras. Otra opción más restrictiva consistiría en considerar como parte del grupo únicamente aquellos genes que pertenezcan en exclusiva a un concepto formal, se miran todos los atributos y se espera que unos estén activos y otros no.

El conocimiento alcanzado mediante esta exploración puede ser interpretado como una exposición, algo que se puede compartir y analizar sin los datos presentes. Además el conocimiento puede ser visualizado, y se han visto varios ejemplos de como realizar esta visualización:

- Retículos cuyos conceptos formales permanecen en posiciones fijas. Así durante la fase de exploración de los distintos valores de φ y ϕ se pueden comprobar entre sí (cfr. Sección 3.4.2).
- Evolución de los genes (objetos) dentro de un retículo de forma que se puede analizar un gen individual y su recorrido por los diferentes conceptos formales a medida que cambia el umbral. Esto permite dar una idea de qué características presenta el gen en función de las diferentes muestras (cfr. Sección 3.4.3).
- Evolución de un concepto formal a medida que cambia el umbral. Se puede analizar cómo un concepto formal cambia en su número de genes (cfr. Sección 3.4.4).

Además se puede considerar que un retículo de conceptos es un índice a contenidos de diferentes fuentes de conocimiento, como ontologías de genes (cfr. Sección 3.7).

Pese a que en la bibliografía no hay una clara formalización de un proceso iterativo EDA basado en FCA hemos tratado de entrever un posible conjunto de pasos que puede llevar a completar un análisis de este tipo. Nos hemos centrado en matrices de expresión genética porque proporcionan un marco con grandes posibilidades: hay muchos datos disponibles para analizar y además hay una gran cantidad de bases de datos con las que se puede enriquecer la información proporcionada por los retículos.

Esta fase de estudio EDA nos dará una visión global de cómo se distribuyen los datos dentros de la GED, pero una vez finalizada es posible continuar con una fase CDA que se centre en áreas más específicas, basándose en datos extraidos por la exploracíon $\mathcal{K}$-FCA, y mediante una posible confirmación empírica.

3.8.1 Una propuesta de metodología de análisis exploratorio de GED con $\mathcal{K}$-FCA

Para mostrar estas ideas y todas las opciones de análisis se ha desarrollado una herramienta disponible vía web(cfr. Apéndice C) y en esta sección queremos presentar un esquema del método para usar el $\mathcal{K}$-FCA para el análisis de datos de expresión:

- **Contextualización.** En esta fase se construye el contexto formal que va a ser analizado, describiendo el conjunto de genes en cuestión, que en gran medida dependen del aparato que se use para la expresión, y, más importante, una descripción de las muestras que se van a usar en el análisis
- **Preparación de los datos.** En esta fase se describe cualquier procesado sobre los datos en crudo previa a la exploración, como el pre-procesado de las medidas de las sondas en el caso de muestras de microarray, etc., según la Sección 3.2.

- **Exploración: evolución del número de conceptos.** Se calcula la gráfica de la evolución del número de conceptos, como se describe en la Sección 3.4.1, y se acotan las zonas interesantes para la exploración.

- **Exploración: co-agrupamiento de genes.** La interfaz gráfica de WebGeneKFCA proporciona una herramienta interactiva para análisis y decisión, permite al usuario navegar a lo largo de diferentes umbrales. De esta forma, utilizando el algoritmo de dibujo de retículos explicado en la Sección 3.4.2, se pueden comparar retículos para diferentes umbrales y ver cómo varían los conceptos formales. En general, realizamos dos tipos de sub-análisis en este punto:

  - **Análisis de la infra-expresión.** Como se dijo antes, la infra-expresión puede ser observada para diferentes valores de existencia en el rango $\varphi \in (-\infty, 0\}$, para lo que se usa el $(\overline{\mathbb{R}}_{\text{máx},+})$-FCA. En puridad, es necesario acotar el rango de exploración con el paso anterior.
  - **Análisis de la sobre-expresión.** El análisis de sobre-expresión es llevado realizado mediante $(\overline{\mathbb{R}}_{\text{mín},+})$-FCApara $\phi \in [0, \infty]$. Se invierte el orden de exploración con respecto al caso $\overline{\mathbb{R}}_{\text{máx},+}$: se comienza en el máximo valor de ϕ y acaba en 0.

  Estos análisis muestran puntos de vista diferentes: no todos los genes no-infraexpresados tienen por qué estar sobre-expresados, o viceversa, para φ y ϕ diferentes. Sólo en el caso en que ambos umbrales sean zero, tendremos informaciones complementarias.

- **Análisis de grupos de genes.** Una vez identificados los grupos de interés tras realizar la exploración se procede a realizar un análisis más detallado de cada uno de los grupos identificados. Siguiendo con el análisis visual descrito en la Sección 3.4.4, por ejemplo, es posible observar la evolución de un grupo de genes a medida que cambia el valor del umbral ϕ/φ.

  En este punto la conexión con bases de datos externas permite obtener información adicioinal de cada gen de forma individual, además se puede establecer una medida de calidad del grupo.

  - **Cálculo de "p-valores".** Sabiendo a que término GO pertenece cada gen y utilizando (3.6.6) es posible calcular la probabilidad de que los genes coagrupados hallan acabado juntos por puro azar. Cuanto más bajo sea su p-valor más garantías hay de que esos genes realmente pertenecen a ese grupo.
  - **Análisis de enriquecimiento.** En el análisis enfocado a las bases de datos externas, en especial con GO, se puede ir un paso más allá utilizando análisis FCA dentro de los propios términos definidos por los genes. Es posible obtener más información acerca de los genes y su relación con los términos GO mirando a los genes que aparecen en un grupo determinado y qué términos GO comparten. Esto produce una matriz booleana con filas (objetos) como genes y columnas (atributos) como términos GO, el correspondiente valor del elemento será 1 si un termino GO está asociado a un gen. Esto se puede representar como un retículo, y de esta forma se pueden confirmar relaciones entre términos GO ya conocidas o incluso sugerir nuevos tipos de relaciones hasta entonces desconocidas.

- **Análisis de algún gen en particular.** A veces es necesario centrarse en un gen en particular para conocer con un mayor detalle su función. En este caso será posible ver a que grupos podría pertenecer con mayor o menor incertidumbre y así ahondar más en su papel regulador dentro de la célula.

Los próximos capítulos presentan casos de estudio con datos simulados y reales, lo que, esperamos, contribuirá a clarificar estas técnicas.

4

Análisis de expresión genética de datos in silico

4.1 Introducción

A diferencia de otros algoritmos de agrupamiento, la técnica $\mathcal{K}$-FCA no produce una salida única sino una secuencia de retículos de conceptos. Una forma de obtener un conocimiento más exacto del funcionamiento del análisis $\mathcal{K}$-FCA aplicado a expresión genética consiste en crear matrices sintéticas siguiendo en cierta medida los patrones que se obtendrían de forma real en laboratorio.

El problema de trabajar con datos reales es que no existe ninguna forma completamente satisfactoria para medir la calidad de los agrupamientos encontrados hasta la fecha. Por ello en muchas ocasiones se simula su comportamiento, intentando sintetizar una matriz de expresión parecida a la que se obtendría con datos reales pero generada a partir de un agrupamiento conocido que pueda ser usado como "supervisión" para evaluar los resultados del co-agrupamiento. Por ese motivo en la Sección 4.2 se definen una serie de grupos que serán utilizados a lo largo de todo el Capítulo. Es común referirse a estos datos como *in silico* dado que no se obtienen de una célula viva sino de una simulación por ordenador.

A la hora de generar una matriz de expresión genética *in silico* hay que tener en cuenta cómo son los datos reales que pretendemos imitar. Tradicionalmente, la técnica más usada para obtener los niveles de expresión genética de diferentes tejidos consistía en utilizar microarrays pero últimamente las técnicas basadas en RNA-Seq están ganando popularidad (véase el Apéndice A). En la Sección 4.3 se entrará más en detalle sobre cómo generar matrices de expresión que imiten las obtenidas en laboratorio.

En la Sección 4.4 expondremos la dificultad de obtener un valor único de φ o ϕ que proporcione todos los grupos con mejor calidad. Se tratará de mostrar la capacidad de agrupar genes para cada una de las diferentes matrices *in silico* generadas en función del umbral. En la Sección 4.5 se muestra cómo diferentes tipos de normalización podrían influir a la hora de generar los coagrupamientos. Por último en la Sección 4.6 se hace una comparación con diferentes algoritmos de coagrupamiento.

4.2 Definiciones de agrupamientos y medidas de desempeño

Con el fin de homogeneizar todas estas comprobaciones se partirá de un conjunto de grupos predefinidos que se adaptarán para cada uno de los diferentes algoritmos. La idea general consiste en generar una matriz $M \in \mathcal{R}^{p \times n}$ donde aparezcan 7 grupos, la matriz tendrá $p = 75{,}000$ filas y $n = 8$ columnas. Cada fila representa un gen y cada columna una muestra diferente. Dentro de la matriz los elementos M_{ij} que pertenecen a cada grupo aparecen en la Tabla 4.1.

Grupo	filas	columnas
A	[0, 4999]	[0, 1]
B	[5000, 9999]	[2, 3]
C	[10000, 24999]	[0, 3]
D	[25000, 29999]	[4, 5]
E	[30000, 34999]	[6, 7]
F	[35000, 49999]	[4, 7]
G	[50000, 69999]	[2, 5]

Tabla 4.1: Dimensiones y localización de los grupos propuestos.

A esta matriz M, tras un procesado, se le aplicará el umbral φ o ϕ en función de si se está tratando de detectar los genes infra- o sobre-expresados. Esto la convertirá en una matriz I^{φ} booleana de donde se obtendrán los coagrupamientos de interés utilizando FCA. En este primer momento hablamos de grupos sin intersecciones, pero a continuación se expondrán los grupos obtenidos a partir de conceptos formales donde sí podrá haber intersecciones y aparecerán las relaciones jerárquicas propias de FCA.

En la Figura 4.1a aparece un esquema de esta matriz M junto con los grupos generados. Pero evidentemente al utilizar *conceptos formales* como grupos y al compartir estos atributos los coagrupamientos serán diferentes de los dados en la Tabla 4.1. Así por ejemplo el *concepto formal* donde se encuentran los genes que afectan las muestras de las columnas $\{0, 1\}$, siguiendo el teorema fundamental de FCA 1.3.4, serán:

$$\mu(\{0, 1\}) = (\{[0, 4999] \cup [10000, 24999]\}, \{0, 1\})$$

Para el concepto donde se encuentran expresados los genes en las condiciones $\{2, 3\}$ serán:

$$\mu(\{2, 3\}) = (\{[5000, 24999] \cup [50000, 69999]\}, \{2, 3\})$$

Las condiciones del concepto $2, 3, 4, 5$ se corresponden con el grupo G porque no comparte todos sus atributos con ningún otro.

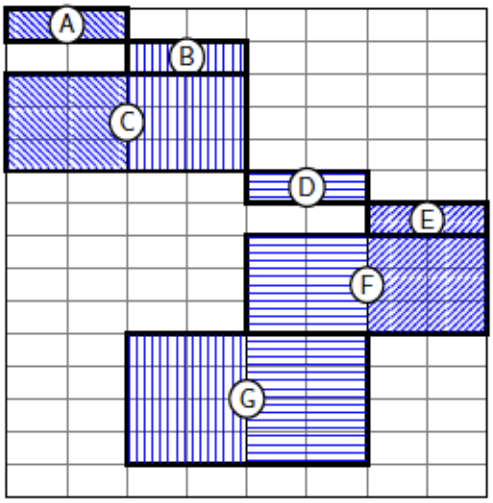
$$\mu(\{2, 3, 4, 5\}) = (\{[50000, 69999]\}, \{2, 3, 4, 5\})$$

Así pues en total de los $2^8 = 256$ posibles *conceptos formales* que podría haber teniendo en cuenta sus columnas nos interesa el valor de los siguientes que se corresponden con los grupos conocidos como se ve en la Tabla 4.2.

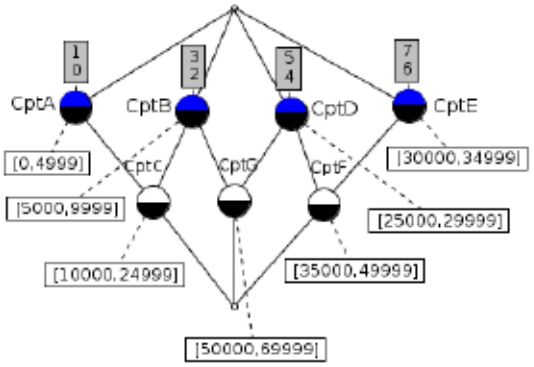
En la Figura 4.1b se muestran estos conceptos ordenados en un retículo.

Nombre	Concepto formal
CptA	$\mu(\{0,1\}) = (\{[0,4999] \cup [10000,24999]\}, \{0,1\})$
CptB	$\mu(\{2,3\}) = (\{[5000,24999] \cup [50000,69999]\}, \{2,3\})$
CptC	$\mu(\{0,1,2,3\}) = (\{[10000,24999]\}, \{0,1,2,3\})$
CptD	$\mu(\{4,5\}) = (\{[25000,29999] \cup [35000,69999]\}, \{4,5\})$
CptE	$\mu(\{6,7\}) = (\{[30000,59999]\}, \{6,7\})$
CptF	$\mu(\{4,5,6,7\}) = (\{[35000,49999]\}, \{4,5,6,7\})$
CptG	$\mu(\{2,3,4,5\}) = (\{[50000,69999]\}, \{2,3,4,5\})$

Tabla 4.2: Dimensiones y localización de los agrupamientos propuestos dados por conceptos formales.



(a) Esquema de los grupos generados.



(b) Retículo que muestra la relación entre los conceptos formales (grupos) junto con el rango de filas y columnas a las que pertenecen.

Figura 4.1: Matriz de expresión genética *in silico*. Simula el nivel de expresión de 8 muestras sometidas a diferentes condiciones.

4.2.1 Matriz de confusión

Para comprobar la calidad de los conceptos encontrados se puede utilizar una matriz de confusión donde la columna muestra el grupo esperado y la fila el grupo clasificado. Las filas y las columnas se identifican con el nombre dado al grupo conocido que se muestra en las Tablas 4.1 y 4.2.

La matriz de confusión para los grupos excluyentes definidos en la Tabla 4.1 debería ser una matriz diagonal en el mejor de los casos, con el valor de la diagonal el número de elementos en ese grupo. En cualquier caso, la suma de toda la fila daría el tamaño del grupo detectado, la suma de la columna daría el tamaño de grupo conocido.

De esta matriz de confusión se puede estimar la probabilidad de clasificar correcta o incorrectamente un gen en un grupo. Así pues estimamos la *probabilidad de detección* definida como la probabilidad de que un gen que pertenece a un grupo finalmente acabe siendo asignado a dicho grupo, contabilizando el complementario de los errores de tipo II o de falso

negativo ([86]). Por otra parte estimamos la *probabilidad de falso positivo* también conocido como *error de tipo I*, que indica lo probable que es que un gen sea erróneamente seleccionado para formar parte de un grupo.

En particular, la *probabilidad de detección* se estima contando el número de clasificaciones correctas del grupo dividido entre el número total de elementos conocidos en el grupo, mirando a la matriz de confusión esto significaría que para el grupo i sería el valor del elemento (i, i) dividido entre la suma de la columna. Cuanto mayor sea su valor mejor clasificado estará un grupo, siendo el máximo valor posible 1 donde todos los genes que pertenecen al grupo se han clasificado correctamente.

Por otro lado, la *probabilidad de falso positivo* se estima contando el número de clasificaciones incorrectas dividido entre todas las clasificaciones realizadas en ese grupo, en la matriz de confusión se calcula para el grupo i sumando todos los elementos de la fila, excepto el elemento (i, i) dividido entre la suma de toda la fila. Cuanto menor sea su valor mejor será la clasificación.

Sin embargo, con el caso de *conceptos formales* (los grupos definidos en la Tabla 4.2) el comportamiento cambia. En esta caso hay que tener en cuenta que los conceptos formales guardan una relación de jerarquía y hay unos conceptos incluidos en otros (ver figura 4.1). Por ese motivo en el caso de una clasificación perfecta el resultado que se obtendría sería como el de la Tabla 4.3. En ella se puede ver que hay 20000 genes en el grupo *CptA*, pero de todos ellos sólo 5000 pertenecen en exclusiva a *CptA*, los 15000 restantes formarán parte del grupo de genes que pertenecen de forma exclusiva a *CptC* $-\mu(\{0, 1, 2, 3\})-$ y por lo tanto también pertenecerán a *CptB* al tener en este caso activas las columnas 2 y 3. En el caso de la segunda línea para *CptB* aparecen 40000 genes en el grupo, pero únicamente 5000 tendrán activas las columnas 2 y 3, dentro de este grupo 15000 genes además tendrán activas las columnas 0 y 1 por lo que pertenecerán también a *CptC* y *CptA*; los otros 20000 restantes pertenecen a *CptG* en exclusiva.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto
CptA	**20000**	15000	15000	0	0	0	0	0
CptB	15000	**40000**	15000	20000	0	0	20000	0
CptC	15000	15000	**15000**	0	0	0	0	0
CptD	0	20000	0	**40000**	15000	15000	20000	0
CptE	0	0	0	15000	**20000**	15000	0	0
CptF	0	0	0	15000	15000	**15000**	0	0
CptG	0	20000	0	20000	0	0	**20000**	0
resto	0	0	0	0	0	0	0	5000

Tabla 4.3: Matriz de confusión para los conceptos formales descritos en la Tabla 4.2 con resolución perfecta.

Esto se puede tratar como una clasificación multi-etiqueta[87, 88] pero no hay una solución consensuada para medir la calidad de estas casos y se deben tomar las medidas con cierta precaución. A la hora de calcular la *probabilidad de detección* y la *probabilidad de falsa alarma*, por ejemplo, hay que tener en cuenta que un gen puede pertenecer a varios grupos a la vez y por lo tanto habrá que restar los genes comunes para realizar el cálculo de probabilidades.

También hay que tener en cuenta que a media que el umbral va disminuyendo para ϕ (aumentando para φ) los genes se van desplazando del concepto supremo al ínfimo. Se

podría llegar a una hipotética situación en la que todos los elementos se encuentren en el concepto ínfimo $\perp$. En este caso, la clasificación evidentemente sería bastante mala, aunque la *probabilidad de detección* sería 1 porque todos los elementos del concepto formal $\perp$ estarían a su vez contenidos en otros conceptos formales. Así, en nuestro ejemplo, todos los elementos de la matriz de confusión tendrían el valor de 75000. Para calcular la *probabilidad de falso positivo* en el caso de CptA por ejemplo habría que calcular cuantos de esos 75000 no pertenecen al grupo y dividirlo entre el número total de elementos asociados al concepto formal, esto es 75000. Se sabe que el concepto formal CptA tiene 20000 elementos, los otros 55000 restantes estarían mal clasificados, por lo tanto su probabilidad de falso positivo es $\frac{55000}{75000}$. Este proceso se realiza con el resto de elementos y se obtiene la probabilidad de falso positivo para todas las agrupaciones de genes en conceptos formales como se muestra en la Tabla 4.4.

Concepto formal	$p_f(\phi), \phi = -\infty$
CptA	$\frac{55000}{75000} \approx 0{,}7333$
CptB	$\frac{35000}{75000} \approx 0{,}4666$
CptC	$\frac{60000}{75000} = 0{,}8$
CptD	$\frac{35000}{75000} \approx 0{,}4666$
CptE	$\frac{55000}{75000} \approx 0{,}7333$
CptF	$\frac{60000}{75000} = 0{,}8$
CptG	$\frac{55000}{75000} \approx 0{,}7333$

Tabla 4.4: Máxima probabilidad de falso positivo posible para un clasificador de los conceptos descritos en la Tabla 4.2

4.2.2 Matriz de similitud de Jaccard.

Otra forma de medir la calidad de los coagrupamientos es mediante la matriz de similitud de Jaccard. Esta medida entre dos conjuntos se define como el cociente entre el número de elementos de la intersección de los conjuntos y el numero de elementos de la unión de los dos conjuntos [69, 89]:

$$J(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|} \tag{4.2.1}$$

Basta con calcular la distancia entre todos los conjuntos detectados y conocidos para obtener la matriz de similitud de Jaccard. Existen diferentes formas de realizar la unión $\cup$ e intersección $\cap$ de dos conjuntos: por filas, por columnas o por ambas como se describirá en detalle a continuación.

Imaginemos una matriz booleana de dimensiones 6×4 como la que se muestra en la figura 4.2. En esta matriz se tienen tres grupos A, B y C. Si se calcula la similitud de Jaccard entre A y C por filas y columnas se tiene:

$$J_{fc}(A, C) = \frac{|A \cap C|}{|A \cup C|} = \frac{|(\{g_2, g_3\}, \{m_0, m_1\})|}{|(\{g_0, g_1, g_2, g_3, g_4, g_5\}, \{m_0, m_1\})|} = \frac{4}{12} \simeq 0{,}3333$$

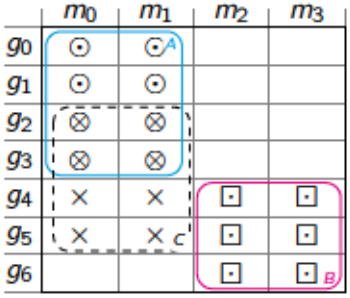


Figura 4.2: Matriz booleana de 6×4. En esta matriz se distinguen tres grupos $A = (\{g_0, g_1, g_2, g_3\}, \{m_0, m_1\})$. $B = (\{g_4, g_5, g_6\}, \{m_2, m_3\})$ y $C = (\{g_2, g_3, g_4, g_5\}, \{m_0, m_1\})$.

Si se hace lo mismo entre B y C se obtiene:

$$J_{fc}(B, C) = \frac{|B \cap C|}{|B \cup C|} = \frac{|\{\emptyset\}|}{|(\{g_4, g_5, g_6\}, \{m_2, m_3\}) \cup (\{g_2, g_3, g_4, g_5\}, \{m_0, m_1\})|} = \frac{0}{14} = 0$$

En cambio si se mira por filas el resultado cambia porque al no tener en cuenta el valor de las columnas los grupos cambiarían. En este caso el valor de similitud entre A y C no cambiaría respecto al caso anterior ya que ambas tienen el mismo número de columnas. Pero entre B y C sí que se encuentran diferencias:

$$J_f(B, C) = \frac{|B \cap C|}{|B \cup C|} = \frac{|\{g_4, g_5\}|}{|\{g_2, g_3, g_4, g_5, g_6\}|} = \frac{2}{5} = 0{,}4$$

De igual forma se podría realizar la misma medida teniendo en cuenta sólo las columnas. Como en esta tesis se está trabajando siempre con coagrupamiento por filas y columnas tiene mucho más sentido realizar la medida de similitud de Jaccard utilizando las filas y las columnas.

La matriz de similitud de Jaccard para los grupos de la Tabla 4.1 es la matriz identidad porque los grupos son excluyentes. En cambio para los grupos descritos por los *conceptos formales* tal y como se describen en la Tabla 4.2 es bastante diferente ya que unos grupos se encuentran incluidos en otros. Por ejemplo para calcular la distancia entre $CptA$ y $CptC$ se parte de que $CptA$ tiene 20000 genes con 2 atributos, de los cuales 15000 pertenecen en exclusiva a $CptC$, de ahí que $|CptA \cap CptC| = 15000 \cdot 2$. En la unión de $|CptA \cup CptC|$ resulta que $CptC$ tiene $15000 \cdot 4$ elementos y $CptA$ tiene $5000 \cdot 2$ elementos en exclusiva que no pertenecen a $CptC$. Por lo tanto:

$$J_{fc}(CptA, CptC) = \frac{|CptA \cap CptC|}{|CptA \cup CptC|} = \frac{15000 \cdot 2}{15000 \cdot 4 + 5000 \cdot 2} = \frac{2}{5} = 0{,}42857$$

Si se repite este proceso para todos los elementos se obtiene la matriz 4.5.

4.3 Generación de matrices in silico

Existen diferentes formas de simular el comportamiento de las matrices de expresión genética (véase [90] para un resumen de diversas técnicas). De todas ellas tal vez la que representa más

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto
CptA	**1**	0	0.42857	0	0	0	0	0
CptB	0	**1**	0.27273	0	0	0	0.33333	0
CptC	0.42857	0.27273	**1**	0	0	0	0	0
CptD	0	0	0	**1**	0	0.27273	0.33333	0
CptE	0	0	0	0	**1**	0.42857	0	0
CptF	0	0	0	0.27273	0.42857	**1**	0	0
CptG	0	0.33333	0	0.33333	0	0	**1**	0
resto	0	0	0	0	0	0	0	1

Tabla 4.5: Matriz de similitud de Jaccard para un clasificador sin errores utilizando los grupos definidos en la Tabla 4.2

fielmente la realidad es la expuesta en [91], que simula todos los parámetros de un microarray, desde la hibridación hasta el ruido que puede aparecer durante el escaneado. Por este motivo, presenta una gran cantidad de parámetros de entrada pero resulta bastante complicada de utilizar sin ofrece una ventaja significativa frente a otros métodos más sencillos.

Así, para comprobar cómo puede afectar el algoritmo de creación de microarrays al agrupamiento de genes, tomamos la decisión de probar con dos algoritmos diferentes utilizados en la literatura. El primero generado con funciones gaussianas de diferente media con la misma varianza, similar al descrito en [92] (Sección 4.3.1) y el segundo, más flexible donde el usuario proporciona una serie de parámetros que gobiernan la generación de los datos [90] (Sección 4.3.2). Más recientemente han aparecido nuevos métodos para medir los niveles de expresión genética basados en nuevas técnicas de secuenciación, también conocidas como *secuenciación de ARN de próxima generación* o RNA-Seq por sus siglas en inglés [93]. La ventaja de este tipo de técnicas radica en su precisión ya que presentan un nivel de ruido mucho menor que las basadas en microarrays. En la Sección 4.3.3 detallamos cómo se puede simular el comportamiento de estos secuenciadores.

4.3.1 Simulación de matrices de microarrays por el método de Ingrid & Speed

Para tratar de simular su comportamiento el primer paso fue analizar el resultado proporcionado por los microarrays comerciales. En concreto se analizó la salida de varios microarrays obtenidos de NCBI. Se descargaron varios ficheros CEL y tras ejecutar la herramienta *apt-probeset-summarize*[1] se obtiene el nivel de expresión de miles de sondas. Esta utilidad, como se explica en el Apéndice A.5.1.1, utiliza el algoritmo RMA[66] para normalizar el ruido de fondo e igualar las distribuciones de las diferentes muestras. En la Figura 4.3 se muestra el nivel de expresión de 100 sondas en diferentes condiciones seleccionadas aleatoriamente.

Los elementos de esa matriz de expresión obtenida a partir de varias muestras pueden obtener valores que van desde $4 \cdot 10^{-5}$ hasta 12300 y se distribuyen tal y como muestra la Figura 4.4a. Si se aplica el logaritmo a cada uno de los valores de expresión genética de la matriz se obtiene la función de densidad de log-probabilidad de la Figura 4.4b donde a simple vista se podrían llegar a intuir que está formada por varias gaussianas de media y varianza diferentes. Ingrid & Speed propusieron en [92] una forma de simular la expresión genética

[1] http://www.affymetrix.com/estore/support/developer/powertools/changelog/apt-probeset-summarize.html.affx

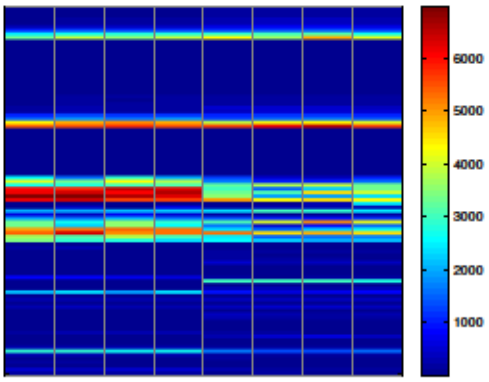
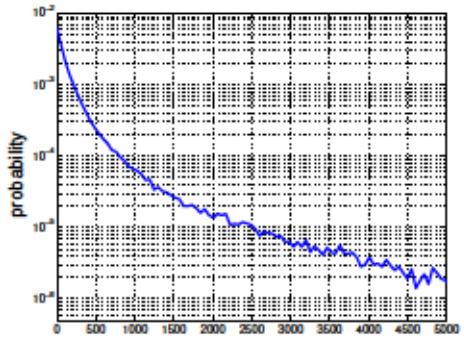
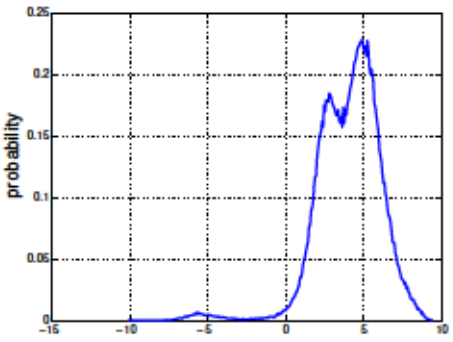


Figura 4.3: Matriz del nivel de expresión de 100 sondas para cada uno de los 8 ficheros CEL

de un microarray y lo hacen suponiendo que la expresión de cada gen es producida por una observación independiente de una distribución normal con una media y varianza que depende del gen y de su nivel de expresión en el experimento dado.



(a) Función de densidad de probabilidad del nivel de expresión.



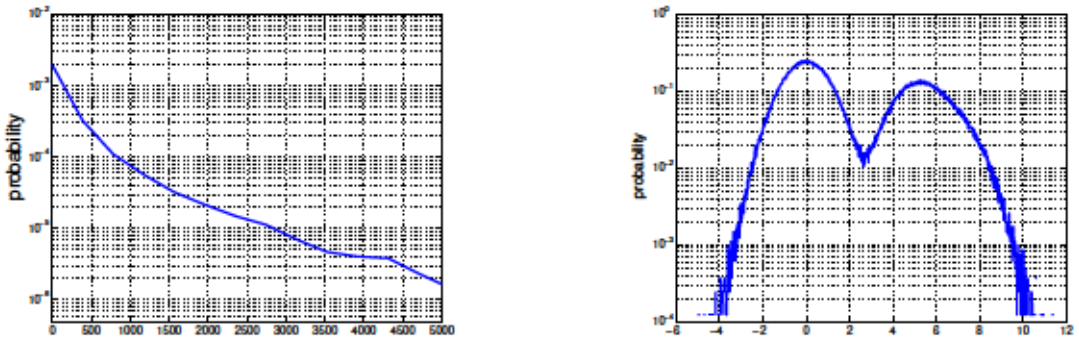
(b) Función de densidad de probabilidad del logaritmo del nivel de expresión.

Figura 4.4: Distribución del nivel de expresión genética que puede presentar cada uno de las sondas de varios experimentos con microarrays.

Debido a su sencillez el primer método de simulación de microarrays que se decidió utilizar fue el basado en funciones gaussianas. El análisis se realizó con una matriz como la explicada en 4.2 de 75000 filas y 8 columnas. Cada fila contiene el nivel de expresión de un gen o sonda, en este nivel de abstracción da igual por lo que consideraremos como si se trataran de genes, para cada una de las 8 diferentes condiciones. En la matriz se definen los 7 grupos principales, aparte en el grupo C y F se han creado tres subdivisiones con 5000 genes cada una, todos ellos tendrán unos niveles de expresión que seguirán una gaussiana de media μ y varianza 1:

Grupo	filas	columnas	μ
A	[0, 4999]	[0, 1]	5
B	[5000, 9999]	[2, 3]	5
C		[0, 3]	
C_a	[10000, 14999]	[0, 3]	5
C_b	[15000, 19999]	[0, 3]	6
C_c	[20000, 24999]	[0, 3]	7
D	[25000, 29999]	[4, 5]	5
E	[30000, 34999]	[6, 7]	5
F		[4, 7]	
F_a	[35000, 39999]	[4, 7]	5
F_b	[40000, 44999]	[4, 7]	6
F_c	[45000, 49999]	[4, 7]	7
G	[50000, 69999]	[2, 5]	5

El resto de los elementos de la matriz que no se encuentran en ningún grupo siguen una distribución normal de media 0 y varianza 1. Una vez que se tiene esta matriz se calcula su exponencial — e^x — por cada una de las celdas. De esta forma los valores de las celdas se distribuirán a lo largo de la curva mostrada en la figura 4.5.



(a) Función de densidad de probabilidad del nivel de expresión.

(b) Función de densidad de probabilidad del logaritmo del nivel de expresión.

Figura 4.5: Función de densidad de probabilidad de expresión de cada celda. Se ha obtenido de una matriz de expresión genética *in silico* creada por gaussianas. Simula el resultado de 8 microarrays con tejidos sometidos a diferentes condiciones, cada microarray mide el nivel de expresión de 75000 genes.

4.3.2 Simulación de matrices de microarrays mediante el método de Dembéle

Otra posible forma de simular el comportamiento de un microarray es el algoritmo descrito por Dembéle en [90] donde se utiliza una combinación de variables que siguen diferentes distribuciones para obtener un resultado que se asemeje lo máximo posible a la salida de un microarray a base de permitir al usuario definir una serie de parámetros y que tiene en cuenta las siguiente características:

- Normalmente hay más genes con baja intensidad que con alta.

- Las (log_2)-intensidades varían entre 0 y 20.
- Bajo condiciones biológicas similares, los niveles de expresión genética varían alrededor un determinado valor medio.
- El número total de gene diferencialmente expresados depende de las condiciones biológicas usadas.
- La variabilidad observada para genes expresados débilmente es mayor que en los fuertemente expresados.

El algoritmo original selecciona de forma aleatoria los genes que aparecerán sobre-expresados, infra-expresados o que no se verán afectados por el experimento. En la modificación aquí propuesta los genes que sí son afectados por el experimento se definen a priori (no se deja al azar). Además puede haber distintas condiciones que afectan la respuesta de los genes y no sólo dos grupos de experimentos. De esta forma se puede genera una matriz con los grupos como los definidos en la Sección 4.2.

El algoritmo se puede resumir en los siguientes pasos, suponiendo que se desea obtener una matriz M con p filas y n columnas:

1. Genera p valores de una distribución beta utilizando las variables a y b como parámetros de entrada, de lo que se obtienen los valores $\overline{x} = \langle x_i, i = 1, ..., p \rangle$
2. Transformar los valores de $\overline{x}$ para que varíen en el rango $[lb, lb + ub] : \overline{z} = lb + ub \cdot \overline{x}$
3. Para cada valor de $\overline{z}$ se generan n valores distribuidos uniformemente en el rango $\mathcal{U}\{((1-\alpha)\overline{z}_i, (1+\alpha)\overline{z}_i)\}$ donde $\alpha = \lambda_1 e^{-\lambda_1 \overline{z}_i}$ y se almacenan en el vector fila $\overline{r}$
   (a) Si el gen i presenta un nivel de expresión diferenciado, es decir, pertenece a un grupo:
      i. si la columna j pertenece al grupo se obtiene una muestra de una distribución normal y se añade a la columna correspondiente del vector $\overline{r}$:
      $\mu_{de} = \mu_{de}^{min} + \{\lambda_2 e^{-\lambda_2}\}$
      $y_{ij} = r_j + \{\mathcal{N}(\mu_{de}, \sigma_{de}^2)\}$
      ii. si la columna j no pertenece al grupo se le asigna el valor correspondiente del vector $\overline{r}$: $y_{ij} = r_j$.
   (b) Si el gen i no pertenece a ningún grupo, se asigna $\overline{r}$ al vector $\overline{y}_i$: $\overline{y}_i = \overline{r}$
4. En este punto se tiene la matriz de expresión genética $Y := \{\overline{y}_i, i = 1, ..., n\rangle$, sin ruido. Se puede añadir ruido a esta matriz creando una matriz de ruido blanco gaussiano $N := \{\mathcal{N}(0, \sigma_n^2), i = 1, ..., n\rangle$. Dejando la matriz con ruido como $X = Y + N$
5. La matriz X posee el logaritmo base 2 de la intensidad de los niveles de expresión. Para conseguir la matriz original hay que elevar cada uno de los elementos por $M_{ij} = 2^{X_{ij}}$

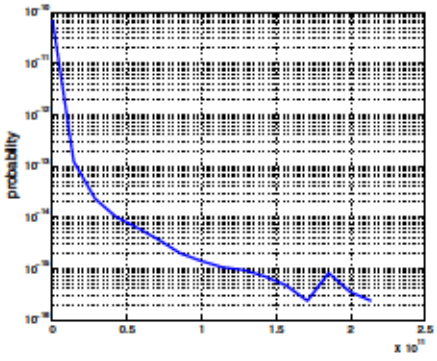
El valor de las constantes se eligió entre los recomendados en [90] y el valor de μ_{de} dependerá del grupo seleccionado como se especifica en la Tabla 4.3.2:

$$a = 2, b = 4, lb = 4, ub = 14, \lambda_1 = 14, \lambda_2 = 2, \sigma_{de} = 2, \sigma_n = 0{,}4$$

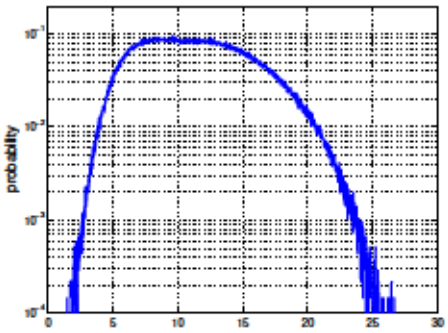
Para comparar los resultados de los dos métodos de simulación de microarray se decidió repetir con la matriz explicada en 4.2 con 75000 filas y 8 columnas donde se definieron los 7 grupos. Lo único que cambia de unos genes a otros es el valor de μ_{de} en función del grupo al que pertenezcan y sólo para las columnas de dentro del grupo:

Grupo	filas	columnas	μ_{de}
A	[0, 4999]	[0, 1]	5
B	[5000, 9999]	[2, 3]	5
C		[0, 3]	
C_a	[10000, 14999]	[0, 3]	5
C_b	[15000, 19999]	[0, 3]	6
C_c	[20000, 24999]	[0, 3]	7
D	[25000, 29999]	[4, 5]	5
E	[30000, 34999]	[6, 7]	5
F		[4, 7]	
F_a	[35000, 39999]	[4, 7]	5
F_b	[40000, 44999]	[4, 7]	6
F_c	[45000, 49999]	[4, 7]	7
G	[50000, 69999]	[2, 5]	5

En la Figura 4.6 se muestra la función de probabilidad que sigue el nivel de expresión para esta matriz.



(a) Función de densidad de probabilidad del nivel de expresión.



(b) Función de densidad de probabilidad del logaritmo del nivel de expresión.

Figura 4.6: Función de densidad de probabilidad del logaritmo del nivel de expresión de cada celda. La matriz de expresión genética *in silico* ha sido creada siguiendo el método descrito por Dembéle. Simula el resultado de 8 microarrays con tejidos sometidos a diferentes condiciones, cada microarray mide el nivel de expresión de 75000 genes.

4.3.3 Simulación de matrices de secuenciación de próxima generación

Las técnicas de secuenciación de próxima generación pueden contar el número de cadenas de ARNm de forma que proporcionan el nivel de expresión con una precisión muy alta (ver el Apéndice A.5.2 para una descripción más detallada de estas técnicas). Debido a la naturaleza discreta del proceso se puede suponer que el nivel de expresión sigue una distribución de Poisson, tal y como sugieren en [44, 94].

La distribución de Poisson muestra la probabilidad de contar x secuencias predefinidas de ARNm en un experimento estimando que el número de moléculas de esa secuencia de ARNm es λ. Así su función de densidad de probabilidad es:

$$f(x) = Pr(X = x) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{4.3.1}$$

Donde el valor de la media y la varianza son iguales a λ.

Con esta función de probabilidad se calcularían los valores de la matriz $N_{gxr} \in \mathbb{R}^{G,X,R}$ donde G es el número total de genes, X es el número de experimentos o tratamientos que se están analizando y R es el número de réplicas que hay en cada experimento. Al final las columnas de experimentos y réplicas se reordenarían para generar una matriz bidimensional $M_{gj} \in \mathbb{R}^{G,X+R}$ con G filas y $X + R$ columnas.

El valor λ se puede suponer que vale:

$$log(\lambda_{gxr}) = s_{gxk} + \alpha_g + \beta_{gx} \tag{4.3.2}$$

Donde s_{gxk} es un factor de normalización, α_g es la expresión media del gen g en los diferentes experimentos y β_{gx} mide el nivel de expresión del gen g en el tratamiento x.

Sin embargo este modelo se puede considerar demasiado simple ya que no hay forma de controlar la media y la varianza por separado. Por eso varios autores sugieren utilizar una función de probabilidad binomial negativa [44, 95, 96]:

$$f(x) = Pr(X = x) = \frac{\Gamma(x + \gamma^{-1})}{\Gamma(\gamma^{-1})\Gamma(x + 1)} \left(\frac{1}{1 + \lambda\gamma}\right)^{\gamma^{-1}} \left(\frac{\lambda}{\gamma^{-1} + \lambda}\right)^x \tag{4.3.3}$$

Donde la media es λ y la varianza $\lambda + \gamma\lambda^2$. La variable γ es el factor de dispersión y se puede comprobar fácilmente como a medida que $\gamma \to 0$ la distribución tiende a una Poisson. En este caso el valor de la media se sigue usando el valor de la fórmula 4.3.4 y como varianza:

$$var(N_{gxr}) = \sigma^2 = \lambda_{gxk} + \gamma_g\lambda^2_{gxk} \tag{4.3.4}$$

Donde γ_g es un parámetro de dispersión asociado a cada gen.

Igual que en los casos anteriores se creará una matriz de expresión genética con 75000 filas y 8 columnas donde se definieron 7 grupos como se explica en 4.2. Lo único que cambia de unos genes a otros es el valor de λ en función del grupo al que pertenezcan y sólo para las columnas de dentro del grupo. El factor de dispersión será igual para cada grupo $\gamma_g = 0{,}4$:

Grupo	filas	columnas	λ
A	[0, 4999]	[0, 1]	1000
B	[5000, 9999]	[2, 3]	1000
C		[0, 3]	
C_a	[10000, 14999]	[0, 3]	1000
C_b	[15000, 19999]	[0, 3]	1200
C_c	[20000, 24999]	[0, 3]	1400
D	[25000, 29999]	[4, 5]	1000
E	[30000, 34999]	[6, 7]	1000
F		[4, 7]	
F_a	[35000, 39999]	[4, 7]	1000
F_b	[40000, 44999]	[4, 7]	1200
F_c	[45000, 49999]	[4, 7]	1400
G	[50000, 69999]	[2, 5]	1000

El resto de los genes que no pertenecen a ningún grupo presentan una media $\lambda = 10$ y un factor de dispersión $\gamma_g = 0{,}6$. En la Figura 4.7 se muestra la función de probabilidad que sigue el nivel de expresión de esta matriz.

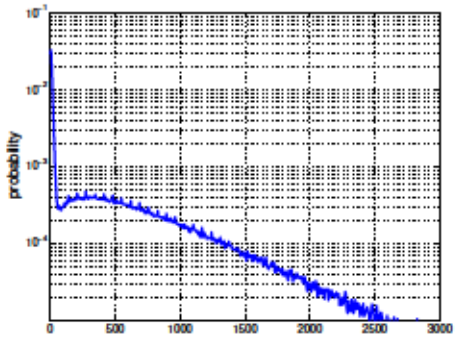


Figura 4.7: Función de densidad de probabilidad del nivel de expresión de cada celda. Matriz de expresión genética *in silico* simulada con una función de probabilidad binomial negativa. Simula el resultado de 8 muestras con tejidos sometidos a diferentes condiciones, para cada tejido se simula el nivel de expresión de 75000 genes.

4.4 Exploración $\mathcal{K}$-FCA de datos in silico

Partiendo de las matrices GED *in silico* creadas en la sección anterior se realizará el análisis explicado en el Capítulo 3. No se realizará un análisis completo utilizando diferentes herramientas visuales como las descritas en la Secciones 3.4.3 y 3.4.4, tampoco se realizará ninguna conexión con sistemas externos al ser genes que no tienen ninguna correspondencia con muestras reales. Este tipo de análisis se ha dejado para el Capítulo 5 donde se muestra todo el potencial que estas técnicas ofrecen con muestras reales. Esta sección nos limitaremos a la exploración e identificación de coagrupamiento de genes. Se obtendrán dos tipos de agrupamientos, unos basados en los genes que pertenecen en exclusiva a un concepto formal y

que podrían corresponder hasta cierto punto con los de la Tabla 4.1 y otros que corresponderán con los conceptos formales completos mostrados en la Tabla 4.2.

La matriz *in silico* M con agrupamientos conocidos es el punto de entrada del sistema de análisis explicado en esta sección. El objetivo es ver cómo varía el nivel de expresión de un gen en función de las diferentes condiciones externas a las que se ve sometido así que el primer paso consiste en una etapa de preprocesado donde la matriz de expresión genética se normaliza por filas.

En la Sección 3.2.2 se explicaron en detalle las diferentes posibilidades que hay a la hora de aplicar un tipo de normalización u otro, además en la Sección 4.5 se explicarán las diferencias entre los distintos métodos propuestos. En esta sección se utilizará una normalización que consiste en dividir todos los elementos de la matriz por la media geométrica de su fila para posteriormente aplicar el logaritmo natural.

Es decir, si M es la matriz de expresión genética, con p filas y n columnas, la normalización consiste en aplicar la fórmula 3.2.2:

$$R_{ij} = log\left(\frac{M_{ij}}{\sqrt[n]{\prod_{j=1}^{n} M_{ij}}}\right) \tag{4.4.1}$$

De esta forma cuando una condición sobre-expresa un gen su valor será positivo y directamente proporcional a su nivel de sobre-expresión en relación con las otras condiciones. De la misma forma, un gen estará más infra-expresado para una condición respecto a otras cuanto menor sea su valor (puesto que siempre es negativo). Sobre esta matriz normalizada R se realizará la exploración $\mathcal{K}$-FCA explicada en la Sección 3.3. Con el propósito de simplificar este ejemplo solamente se analizarán los retículos obtenidos en la exploración en el dominio $\mathbb{R}_{\text{mín},+}$, es decir, se agruparán los genes sobre-expresados.

4.4.1 Exploración de matrices de microarrays de Ingrid & Speed

El primer paso en la exploración de la matriz R consiste en calcular el número de conceptos formales en función de los umbrales ϕ/φ para los dominios $\mathbb{R}_{\text{mín},+}$ y $\mathbb{R}_{\text{máx},+}$ tal y como se explica en la Sección 3.4.1. El resultado se puede observar en la Figura 4.8.

El número de conceptos que aparece durante la exploración nos da una idea de la complejidad de los retículos de conceptos. El dominio $\mathbb{R}_{\text{mín},+}$ muestra los grupos de genes sobre-expresados, la exploración se realiza en el rango $\varphi \in (-\infty, 0]$. Según la Figura 4.8 el número de conceptos se estabiliza en 41 para el rango de $\phi \in [1{,}79, 3{,}45]$, por lo que parece interesante observar los retículos que hay antes, después y en medio de este intervalo. En la Figura 4.9 se muestran cuatro retículos para los valores de $\phi = 3, 5, 2, 2, 1, 5, 0$. Estos retículos se han dibujado siguiendo las reglas descritas en la Sección 3.4.2 de forma que los conceptos formales se pueden comparar para diferentes valores e umbral.

A medida que se va disminuyendo el umbral ϕ los genes se van desplazando desde el concepto supremo – $\top$ – al ínfimo –$\bot$–. En la Figura 4.9a se ven pocos conceptos, alguno de

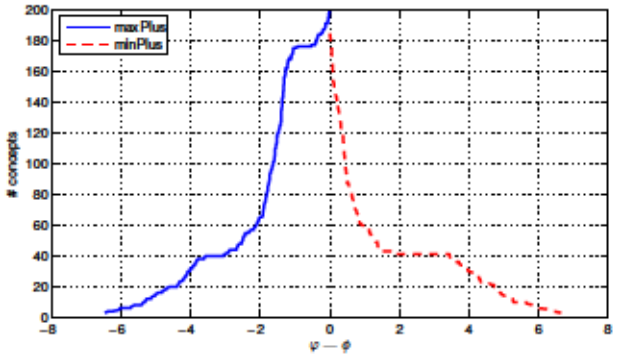


Figura 4.8: Número de conceptos en función del umbral φ y ϕ

los cuales se corresponde con los grupos conocidos como el A, B, D y E. Los otros conceptos que se ven en la figura son conceptos formales que aparecen por efecto del ruido. Según disminuye el valor de ϕ los genes se van desplazando hacia abajo y aparecen nuevos conceptos. En la Figura 4.9b se ven claramente los conceptos formales de los grupos A, B, C, D, E y F. Si se sigue disminuyendo el valor de ϕ aparecen más conceptos asociados con un mayor ruido.

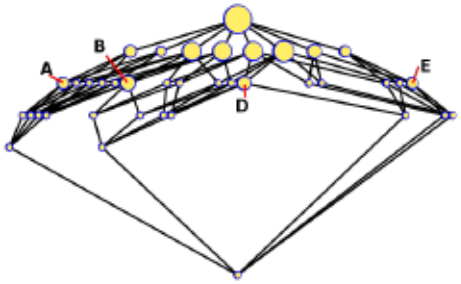
4.4.1.1 Co-agrupamientos basados en ϕ-conceptos formales

Para comprobar el efecto que tiene el umbral ϕ a la hora de definir los co-agrupamientos correctamente se muestra el efecto de dicho umbral en dos tipos de probabilidades, la *probabilidad de detección* y la *probabilidad de falso positivo* como se explicó en la Sección 4.2.1.

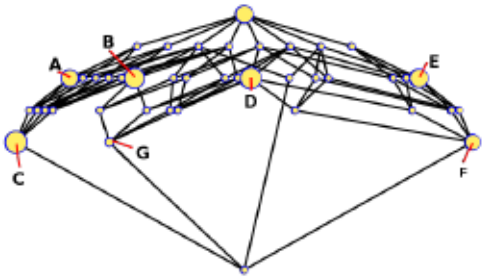
Se tratará de encontrar los co-agrupamientos etiquetados como *CptA*, *CptB*, *CptC*, *CptF* y *CptG* dados por los conceptos formales de la Tabla 4.2. Si la clasificación fuera completamente correcta se tendría que el co-agrupamiento *CptA* es igual al concepto formal dado por $\mu(\{0,1\})$, o que *CptC* es igual al concepto formal dado por $\mu(\{0,1,2,3\})$. Pero como habrá errores de clasificación esta identidad no se llegará a producir y los conceptos formales identificados en los diferentes retículos durante la exploración min-plus coincidirán en mayor o menor medida con lo esperado. Por ese motivo, hemos decidido mostrar las curvas de probabilidad para estos co-agrupamientos.

Así, las curvas de probabilidad en función de ϕ se muestran en la Figura 4.10a para la *probabilidad de detección* y en la Figura 4.10b para la *probabilidad de falso positivo* en los coagrupamientso para los grupos antes descritos. Las curvas de probabilidad de los conceptos formales para CptC y CptF son muy parecidas como cabría esperar porque tienen un número de objetos y genes parecidos. Fácilmente se puede comprobar como a medida que disminuye el valor de ϕ aumenta la probabilidad de detección, es más fácil detectar un gen que pertenece a ese grupo pero a cambio aumenta bastante el ruido lo cual explica que la probabilidad de falso positivo aumente.

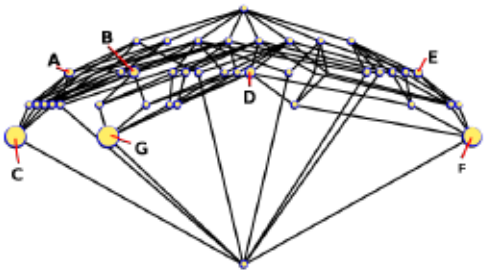
En la Tabla 4.6 se muestra la matriz de confusión para $\phi = 0$ donde la columna muestra el grupo esperado y la fila el grupo clasificado. En este caso se muestran todos los grupos definidos en la Tabla 4.2. En la columna y fila identificada como *resto* se agrupan todos



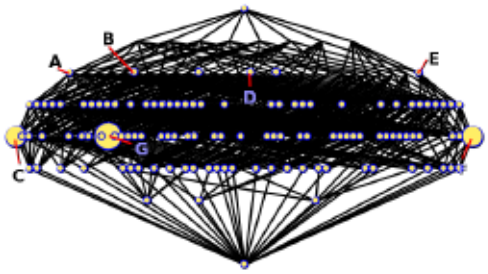
(a) Diagrama de Hasse para $\phi = 3{,}5$



(b) Diagrama de Hasse para $\phi = 2{,}2$

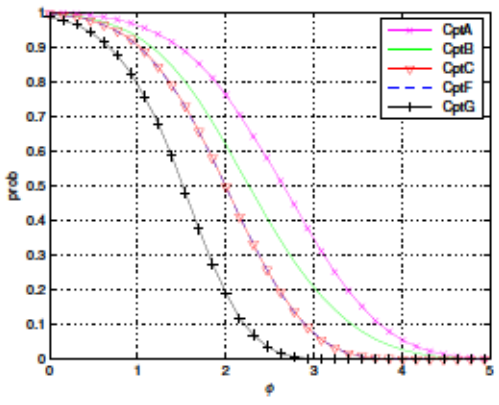


(c) Diagrama de Hasse para $\phi = 1{,}5$

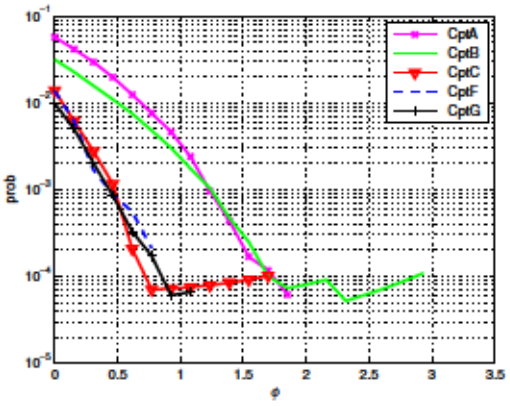


(d) Diagrama de Hasse para $\phi = 0$

Figura 4.9: Retículos de conceptos para diferentes valores de ϕ en el dominio minplus para microarrays simulados mediante el método propuesto por Ingrid & Speed.



(a) Probabilidad asignar correctamente un gen al concepto correspondiente.



(b) Probabilidad de asignar a un concepto un gen erróneo.

Figura 4.10: Prestaciones en función de ϕ para microarrays simulados mediante el método propuesto por Ingrid & Speed.

aquellos genes que no tenían un grupo predefinido, es decir, todos los genes que no encajan en uno de los 7 grupos que se definieron previamente. En la columna de la derecha se muestra la *probabilidad de falso positivo* y en la última fila se muestra la *probabilidad de detección* para cada uno de los grupos.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto	pf
CptA	**19953**	14974	14953	26	29	0	0	1130	0.0570
CptB	14989	**39815**	14967	19876	26	0	19849	1218	0.0315
CptC	14942	14941	**14920**	1	0	0	0	165	0.0138
CptD	18	19866	0	**39799**	14988	14958	19841	1066	0.0278
CptE	21	23	0	14977	**19951**	14953	1	1103	0.0554
CptF	0	1	0	14935	14941	**14911**	1	151	0.0136
CptG	0	19715	0	19717	0	0	**19690**	144	0.0099
resto	0	1	0	0	2	0	0	1341	
pd	0,9977	0,9954	0,9947	0,9950	0,9976	0,9941	0,9845		

Tabla 4.6: Matriz de confusión para $\phi = 0$

A la hora de mirar la matriz de confusión hay que tener en cuenta que los conceptos formales guardan una relación de jerarquía y hay unos conceptos incluidos en otros (ver Figura 4.1). Para aclarar este punto e indicar cómo se ha de leer esta matriz a continuación se expone un ejemplo donde se explican los resultados para el grupo *CptA*. En este *concepto formal* se han detectado 19953 objetos de los 20000 que originalmente debería tener *CptA*, así que la *probabilidad de detección* en este caso es bastante alta, de 0,9977. De los 20000 genes que se esperan en *CptA*, 15000 vienen exclusivamente de *CptC*, en este caso de esos 19953 genes detectados, 14953 pertenecen exclusivamente a *CptC*. A su vez los genes de *CptC* además de pertenecer a *CptC* también pertenecen a *CptB* porque comparten los atributos de las columnas 2 y 3. Por este motivo si no hubiera errores de falsos positivos el número de genes en *CptB* debería ser igual al de *CptC*, pero la realidad es que hay 21 elementos más en *CptB* que en *CptC*. También hay 26 elementos mal clasificados como pertenecientes al grupo *CptD* y 29 a *CptE*, así como 1130 que no corresponden con ninguno de los grupos anteriores. Esto significa que hay $21 + 26 + 29 + 1130 = 1206$ elementos mal clasificados, de un total $19953 + 1206 = 21159$ elementos encontrados para *CptA*, lo que da una *probabilidad de falso positivo* de $\frac{1206}{21159} = 0{,}57$.

Si se mira el valor de la matriz de confusión para un valor más alto como $\phi = 1{,}5$ los resultados que se obtienen son los esperados según la Figura 4.10, esto significa una peor *probabilidad de detección* dentro de los grupos a cambio de una disminución en el número de clasificaciones erróneas. Esta matriz se muestra en la Tabla 4.7.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto	pf
CptA	**17897**	12990	12989	0	0	0	0	3	2.2345e-04
CptB	12930	**32428**	12930	14581	0	0	14581	10	3.0828e-04
CptC	11200	11201	**11200**	0	0	0	0	0	8.9278e-05
CptD	0	14564	0	**32370**	12877	12877	14564	9	2.7796e-04
CptE	0	1	0	12984	**17897**	12983	1	5	3.3514e-04
CptF	0	0	0	11149	11149	**11149**	0	0	0
CptG	0	10155	0	10155	0	0	**10155**	0	0
resto	373	1373	281	1369	375	289	1009	4973	
pd	0,8949	0,8107	0,7467	0,8093	0,8949	0,7433	0,5078		

Tabla 4.7: Matriz de confusión para $\phi = 1{,}5$

Otra forma de medir la calidad de los coagrupamientos es mediante la matriz de similitud de Jaccard como ya se explicó en 4.2.2. En la Tabla 4.8 se muestra la matriz de similitud de Jaccard para $\phi = 0$ y en la Tabla 4.9 para $\phi = 1{,}5$.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto
CptA	**0.94091**	0	0.42309	0	0	0	0	0
CptB	0	**0.96421**	0.2701	0	0	0	0.32827	2.2039e-05
CptC	0.413	0.2666	**0.981**	0	0	0	0	0
CptD	0	0	0	**0.96743**	0	0.27009	0.32832	0
CptE	0	0	0	0	**0.94242**	0.42337	0	7.8821e-05
CptF	0	0	0	0.2672	0.41343	**0.98066**	0	0
CptG	0	0.32402	0	0.32475	0	1.4239e-05	**0.97495**	0
resto	0.02823	0.020337	0.0066095	0.017805	0.027563	0.0060485	0.0048416	0.26804

Tabla 4.8: Matriz de similitud de Jaccard para $\phi = 0$

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto
CptA	**0.89467**	0	0.35895	0	0	0	0	0.0079087
CptB	0	**0.8105**	0.21877	0	0	0	0.20247	0.020752
CptC	0.37205	0.26117	**0.74662**	0	0	0	0	0.0098641
CptD	0	0	0	**0.80907**	0	0.21797	0.20247	0.02069
CptE	0	0	0	0	**0.89458**	0.35792	0	0.0079515
CptF	0	0	0	0.26013	0.37179	**0.74327**	0	0.010148
CptG	0	0.25202	0	0.25191	0	0	**0.50775**	0.030801
resto	7.916e-05	0.00019074	0	0.00017185	0.00013193	0	0	0.71958

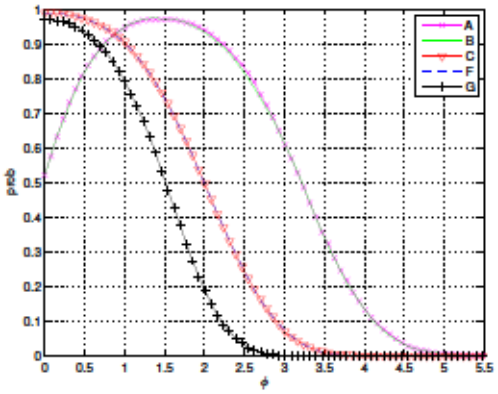
Tabla 4.9: Matriz de similitud de Jaccard para $\phi = 1{,}5$

Según disminuye el valor de ϕ la *probabilidad de detección* aumenta lógicamente porque cada vez más genes se van moviendo desde el concepto supremo que no tiene ningún atributo hasta el concepto ínfimo ($\perp$) que tiene todos. La exploración se realiza en el intervalo $\phi \in [0, \infty)$ porque el rango $(-\infty, 0]$ será explorado por $\overline{\mathbb{R}}_{\text{máx},+}$. Si se siguiera bajando el valor de ϕ hasta llegar a $-\infty$ el valor de la *probabilidad de falso positivo* tendería al valor indicado en la Tabla 4.2. Pero a veces no interesa obtener una lista de genes que pertenecen a un concepto dado, sino la lista de genes que se encuentran activados o desactivados específicamente en ciertas condiciones. Esto se explica en la siguiente sección.

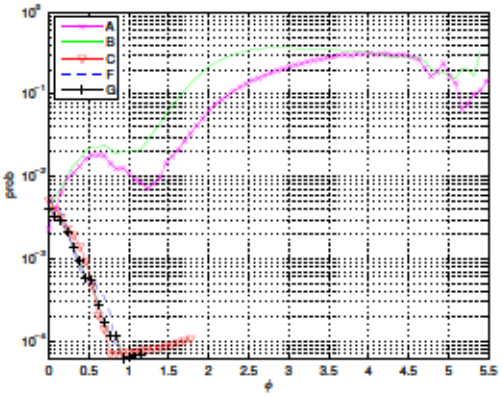
4.4.1.2 Co-agrupamientos basados en ϕ-conceptos con genes propios

También es posible considerar como un grupo no todo el *concepto formal* como se ha hecho en el apartado anterior sino sólo los genes que exclusivamente pertenecen a ese concepto, sin tener ningún otro atributo activo. Es decir, ahora se buscarán los grupos A-G disjuntos tal y como se describen en la Tabla 4.1. Hemos repetido las medidas partiendo de esta premisa y las *probabilidades de detección* aparecen en las Figuras 4.11a y 4.11b para la *probabilidad de falso positivo* en los grupos A, B, C, F y G. Las curvas de probabilidad de los grupos A y B por una parte y C y F son similares porque los grupos también son parecidos (presentan distribuciones similares).

Cada conjunto de grupos parece mostrar un comportamiento diferente. Para los grupos que comprenden pocas muestras (es decir, columnas de la matriz de expresión) como por ejemplo los grupos A y B (sólo dos columnas de la matriz de expresión genética) la probabilidad



(a) Probabilidad asignar correctamente un gen al grupo dado.



(b) Probabilidad de asignar a un grupo un gen que no pertenece a ese grupo.

Figura 4.11: Prestaciones en función de ϕ para microarrays simulados mediante el método propuesto por Ingrid & Speed.

de detección aumenta a medida que el valor de ϕ disminuye hasta llegar a un valor entorno a $\phi \approx 1{,}5$ y a partir de ese valor la probabilidad de detección disminuye. Ese punto en torno a $\phi \approx 1{,}5$ coincide con el crecimiento del número de conceptos formales que aparecen en la Figura 4.8. En cambio para los grupos C, F y G, dicho máximo de probabilidad de detección y de número de elementos en el grupo se encuentra en $\phi = 0$.

Para los grupos A y B según disminuye el valor de ϕ la probabilidad de falso positivo tiende a disminuir hasta llegar a un mínimo en torno a $\phi \approx 1{,}2$ en el grupo A y $\phi \approx 1$ en el grupo B para después seguir disminuyendo. La probabilidad de falso positivo siempre es mayor para el grupo B que para el A porque algunos elementos pertenecientes el grupo B erróneamente se pueden clasificar dentro del grupo G con el que comparte dos columnas. En cambio para los grupos C, F y G la probabilidad de falso positivo parece aumentar según se acerca al valor $\phi = 0$.

En la Tabla 4.10 se muestra la matriz de confusión para $\phi = 1{,}5$ donde la columna muestra el grupo esperado y la fila el grupo clasificado. En la columna y fila identificada como *resto* se agrupan todos aquellos genes que no tenían un grupo predefinido, es decir, todos los genes que no encajan en uno de los 7 grupos que se definieron previamente. En la columna de la derecha se muestra la *probabilidad de falso positivo* y en la última fila se muestra la *probabilidad de detección* para cada uno de los grupos. Tal y como cabría esperar al observar la Figura 4.11 la *probabilidad de falso positivo* es bastante baja, en cambio la *probabilidad de detección* es alta para los grupos A, B, D y E.

Si se baja el umbral hasta $\phi = 0$ la matriz de confusión que aparece es la que se muestra en la Tabla 4.11 donde se ve como la probabilidad de detección ha aumentado considerablemente para los grupos C, F y G a costa de disminuir la probabilidad de detectar genes en los grupos A, B, D y E. Aquí también se puede calcular la matriz de similitud de Jaccard, que para $\phi = 0$ se muestra en la Tabla 4.12.

	A	B	C	D	E	F	G	resto	pf
A	**4853**	0	71	0	0	0	0	3	$\frac{74}{4927} = 0{,}015$
B	0	**4866**	56	0	0	0	266	9	$\frac{331}{5197} = 0{,}0637$
C	0	1	**11200**	0	0	0	0	0	$\frac{1}{11201} = 8{,}9 \cdot 10^{-5}$
D	0	0	0	**4885**	0	57	252	8	$\frac{317}{5202} = 0{,}0609$
E	0	0	0	0	**4878**	73	1	5	$\frac{79}{4957} = 0{,}0159$
F	0	0	0	0	0	**11149**	0	0	$\frac{0}{11149} = 0$
G	0	0	0	0	0	0	**10154**	0	$\frac{0}{10154} = 0$
resto	147	133	3673	115	122	3721	9327	4975	
pd	$\frac{4853}{5000}$ 0,97	$\frac{4866}{5000}$ 0,973	$\frac{11200}{15000}$ 0,747	$\frac{4885}{5000}$ 0,977	$\frac{4878}{5000}$ 0,976	$\frac{11149}{15000}$ 0,743	$\frac{10154}{20000}$ 0,508		

Tabla 4.10: Matriz de confusión para $\phi = 1{,}5$

	A	B	C	D	E	F	G	resto	pf
A	**2609**	0	0	0	0	0	0	6	$\frac{6}{2615} = 0{,}00229$
B	0	**2603**	0	0	0	0	0	9	$\frac{9}{2612} = 0{,}00344$
C	21	19	**14843**	0	0	0	0	37	$\frac{77}{14920} = 0{,}00516$
D	0	0	0	**2580**	0	0	0	7	$\frac{7}{2587} = 0{,}00271$
E	0	0	0	0	**2629**	0	0	4	$\frac{4}{2633} = 0{,}00152$
F	0	0	0	23	27	**14826**	0	33	$\frac{83}{14909} = 0{,}00557$
G	0	21	0	25	0	0	**19392**	34	$\frac{80}{19472} = 0{,}00411$
resto	2370	2357	157	2372	2344	174	608	4870	
pd	$\frac{2609}{5000}$ 0,522	$\frac{2603}{5000}$ 0,521	$\frac{14843}{15000}$ 0,99	$\frac{2580}{5000}$ 0,52	$\frac{2629}{5000}$ 0,526	$\frac{14826}{15000}$ 0,988	$\frac{19392}{20000}$ 0,97		

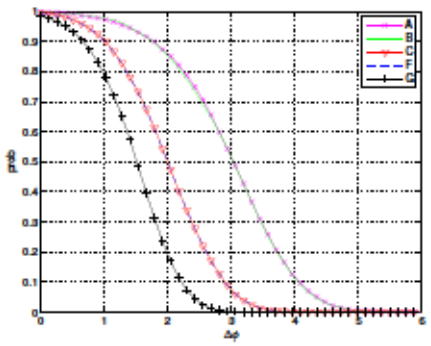
Tabla 4.11: Matriz de confusión para $\phi = 0$

ϕ versus $\Delta\phi$ para co-agrupamientos basados en ϕ-conceptos con genes propios Cada grupo parece tener unas características diferentes y por ello el valor de ϕ óptimo para detectar cada grupo varía. El valor del umbral ϕ no parece muy adecuado para ser usado directamente como indicador del grado de confianza con el que un gen pertenece de forma exclusiva a un determinado grupo porque las probabilidades presentan máximos en diferentes valores en función del grupo. Interesa encontrar otra variable según la cual las funciones *probabilidad de detección* y *probabilidad de falso positivo* sean monótonas de forma que a medida que aumente (o disminuya) esa variable también lo hagan las probabilidades. Esta variable podría ser $\Delta\phi$ que se define como el rango de valores de ϕ por el cual un objeto pertenece exclusivamente a un concepto, o lo que es lo mismo, un gen pertenece a un grupo. En el caso minplus cuanto mayor sea $\Delta\phi$ mayor será la diferencia de expresión entre los casos para los que un gen se espera sobre-expresado y el resto. Así pues a mayor $\Delta\phi$ la *probabilidad de falso positivo* baja a costa de también bajar la *probabilidad de detección* .

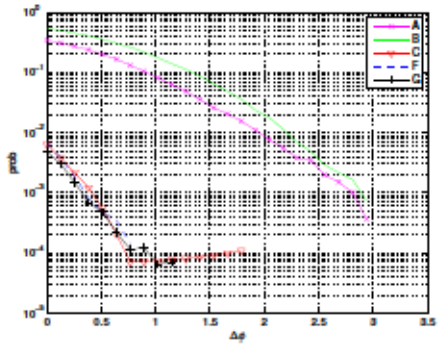
En la Figura 4.12 se muestran las curvas de probabilidad para los grupos A,B,C,F y G en función de $\Delta\phi$. Si se comparan estas curvas con las de la Figura 4.11 se observa cómo la probabilidad de detección aumenta a costa de aumentar también la probabilidad de falso positivo, la cual llega a ser bastante alta, en torno a 0,42 para el grupo A y $\Delta\phi = 0$.

	A	B	C	D	E	F	G	resto
A	**0.521**	0	0.0006	0	0	0	0	0.0372
B	0	**0.512**	0.00054	0	0	0	0.0005	0.037
C	0	0	**0.984**	0	0	0	0	0.00346
D	0	0	0	**0.515**	0	0.0007	0.0006	0.037
E	0	0	0	0	**0.525**	0.0008	0	0.037
F	0	0	0	0	0	**0.983**	0	0.0038
G	0	0	0	0	0	0	**0.966**	0.0122
resto	0.0003	0.0004	0.001	0.0003	0.0002	0.001	0.0012	0.317

Tabla 4.12: Matriz de Jaccard para $\phi = 0$



(a) Probabilidad asignar correctamente un gen al grupo dado.



(b) Probabilidad de asignar a un grupo un gen que no pertenece a ese grupo.

Figura 4.12: Prestaciones en función de $\Delta\phi$ para microarrays simulados mediante el método propuesto por Ingrid & Speed.

4.4.2 Exploración de matrices de microarrays de Dembéle

Al igual que se hizo anteriormente esta matriz de expresión *in silico* normalizada, R, con grupos previamente conocidos es el punto de entrada de la exploración $\mathcal{K}$-FCA. La exploración en $\varphi \in (-\infty, 0]$ y $\phi \in [0, \infty)$ permite ver como evoluciona el número de conceptos tal y como muestra la Figura 4.13. Se puede comprobar que la evolución del número de conceptos es muy parecida a la observada en la Figura 4.8 para el caso anterior.

Para observar los genes sobre-expresados se analiza el dominio minplus, donde al igual que en el caso anterior, se pueden sacar cientos de retículos diferentes en función del valor de ϕ. La forma de los retículos es similar a que se ve en las Figura 4.9a-4.9d ya que la estructura de los agrupamientos es muy parecida, la forma de generar los datos parece tener poco efecto en los retículos de conceptos generados.

4.4.2.1 Co-agrupamientos basados en ϕ-conceptos formales

Conviene estudiar en detalle cómo afecta ϕ a las probabilidades de definir las agrupaciones correctamente. Las curvas de probabilidad se muestran en las Figuras 4.14 donde se puede

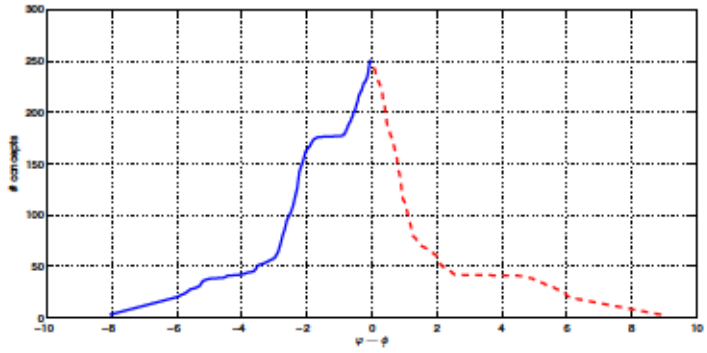
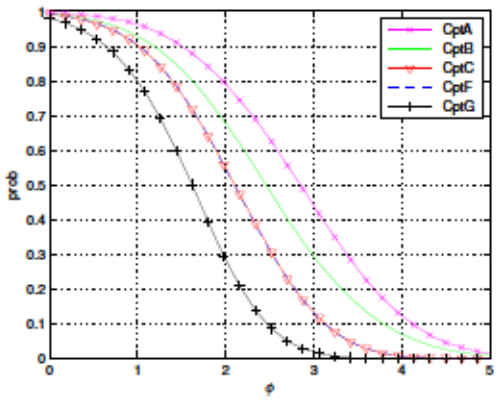
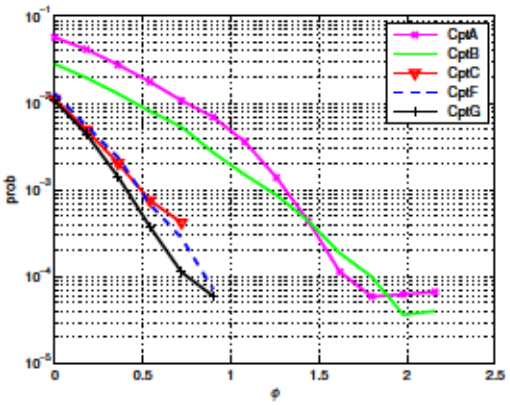


Figura 4.13: Número de conceptos en función del umbral φ y ϕ para una matriz *in silico* generada mediante el método propuesto por Dembéle.

ver lo parecidas que son a las curvas de las Figuras 4.10.



(a) Probabilidad asignar correctamente un gen al concepto correspondiente.



(b) Probabilidad de asignar a un concepto un gen erróneo.

Figura 4.14: Prestaciones en función de ϕ para microarrays simulados mediante el método propuesto por Dembéle.

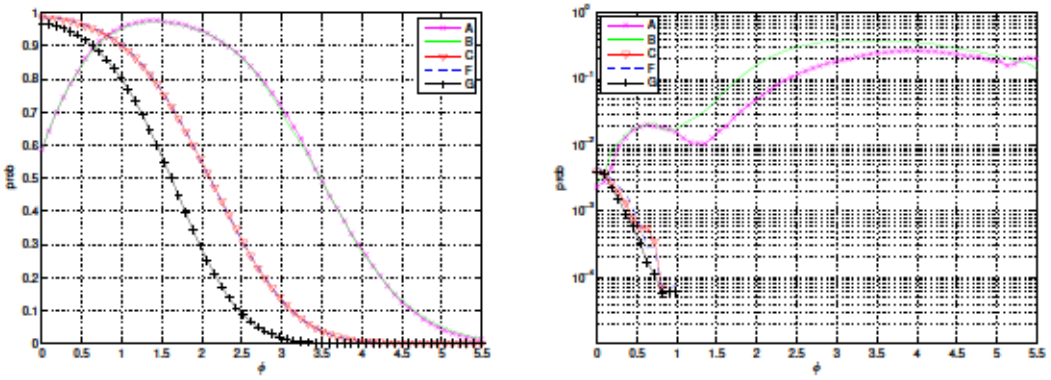
La matriz de similitud de Jaccard para $\phi = 0$ se muestra en 4.13 y guarda bastante parecido con la matriz de la sección anterior mostrada en la Tabla 4.8.

4.4.2.2 Co-agrupamientos basados en ϕ-conceptos con genes propios

En el caso de considerar exclusivamente aquellos genes que únicamente tienen los atributos deseados, no el concepto formal entero, las curvas de probabilidad también guardan una gran similitud respecto a las de la sección anterior. Las curvas de probabilidad se muestran en las Figuras 4.15 donde se puede ver lo parecidas que son a las curvas de las Figuras 4.11.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto
CptA	**0.94032**	0	0.42318	0	0	0	0	0
CptB	0	**0.96552**	0.26994	0	0	0	0.3268	0
CptC	0.41275	0.26694	**0.98121**	0	0	0	0	0
CptD	0	0	0	**0.96177**	0	0.26951	0.32709	0
CptE	0	0	0	0	**0.94287**	0.42271	0	3.9747e-05
CptF	0	0	0	0.26607	0.41354	**0.97934**	0	0
CptG	0	0.3241	0	0.32333	0	0	**0.96972**	0
resto	0.028743	0.018363	0.0054581	0.020842	0.028106	0.0065428	0.0058377	0.25775

Tabla 4.13: Matriz de similitud de Jaccard para $\phi = 0$



(a) Probabilidad asignar correctamente un gen al grupo dado.

(b) Probabilidad de asignar a un grupo un gen que no pertenece a ese grupo.

Figura 4.15: Prestaciones en función de ϕ para la matriz *in silico* creada mediante el método propuesto por Dembéle.

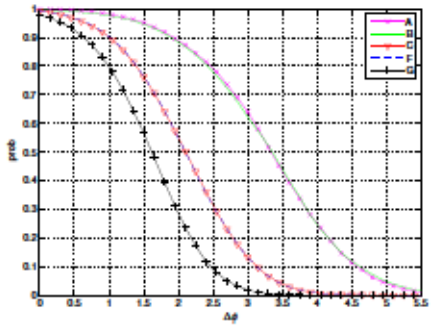
ϕ versus $\Delta\phi$ para co-agrupamientos basados en ϕ-conceptos con genes propios La gráficas para $\Delta\phi$ definida como el rango de valores el rango de valores de ϕ por el cual un gen pertenece exclusivamente a un grupo, tal y como se definió para la figura 4.12, también se muestran en 4.16. Se puede ver que las dos Figuras son muy parecidas lo que parece confirmar una vez más que el método para generar la matriz de expresión genética tiene un escaso efecto en los retículos de conceptos.

4.4.3 Exploración de matrices de secuenciación de próxima generación

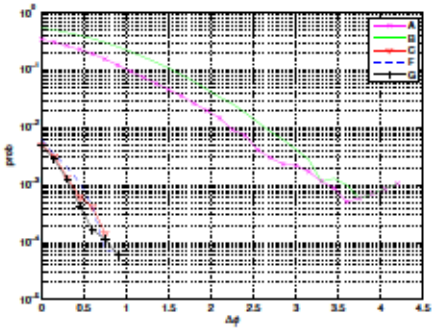
En esta sección se va a estudiar la matriz *in silico R* generada en la Sección 4.3.3 siguiendo una distribución binomial negativa. El preprocesado que se aplicará es el mismo que en ocasiones anteriores y viene dado por la ecuación 4.4.1.

Esta matriz normalizada se utiliza para realizar la exploración $\mathcal{K}$-FCA. La exploración en $\varphi \in (-\infty, 0]$ y $\phi \in [0, \infty)$ permite ver como evoluciona el número de conceptos tal y como muestra la Figura 4.17.

Para observar los genes sobre-expresados se analiza el dominio minplus, donde al igual



(a) Probabilidad asignar correctamente un gen al grupo dado.



(b) Probabilidad de asignar a un grupo un gen que no pertenece a ese grupo.

Figura 4.16: Prestaciones en función de $\Delta\phi$ para la matriz *in silico* creada mediante el método propuesto por Dembéle.

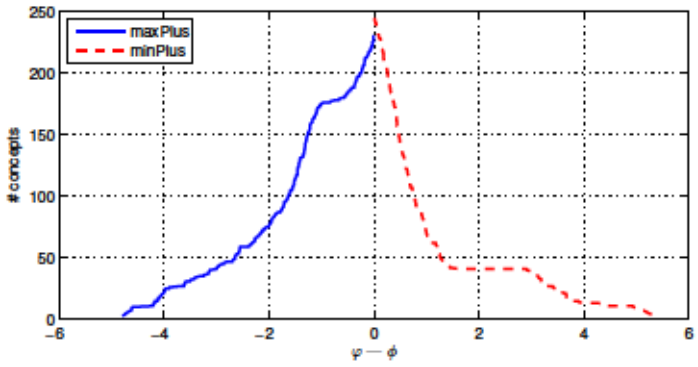
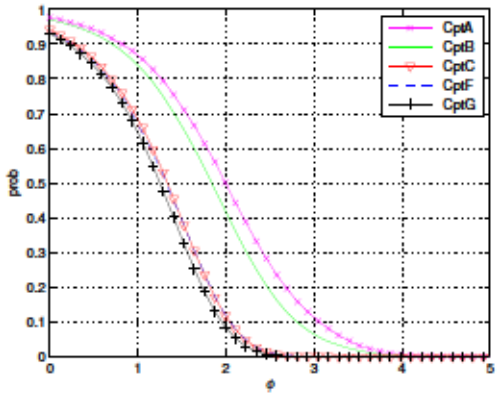


Figura 4.17: Número de conceptos en función del umbral φ y ϕ para una matriz *in silico* generada siguiendo una distribución binomial negativa.

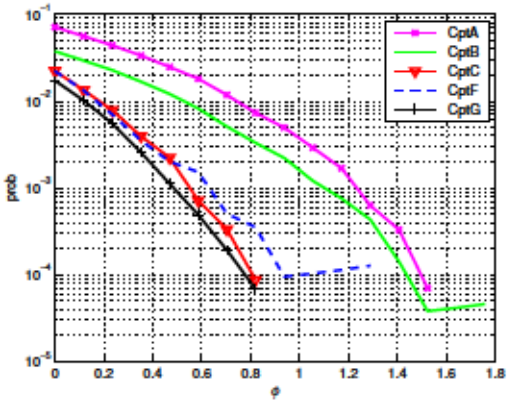
que en la Sección 4.3.1, se pueden sacar cientos de retículos diferentes en función del valor de ϕ. Aunque la función de probabilidad de esta matriz de expresión *in silico* es bastante diferente de la creada en las secciones anteriores los retículos de conceptos formados son bastante parecidos, su forma es como la que se ve en las Figuras 4.9a-4.9d.

4.4.3.1 Co-agrupamientos basados en ϕ-conceptos formales

Al igual que en las secciones anteriores es necesario estudiar cómo afecta ϕ a las probabilidades de definir los grupos correctamente. Las curvas de probabilidad se muestran en las Figuras 4.18 donde se puede ver lo parecidas que son a las curvas de las Figuras 4.10.



(a) Probabilidad asignar correctamente un gen al concepto correspondiente.



(b) Probabilidad de asignar a un concepto un gen erróneo.

Figura 4.18: Prestaciones en función de ϕ para una matriz *in silico* generada siguiendo una distribución binomial negativa.

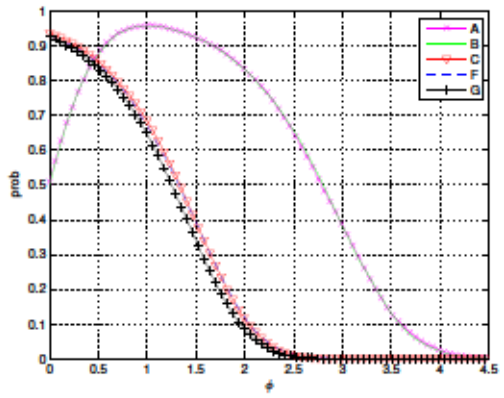
La matriz de similitud de Jaccard para $\phi = 0$ se muestra en la Tabla 4.14 y guarda bastante parecido con las generadas con los otros métodos en 4.8 y 4.13.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG	resto
CptA	**0.90833**	0	0.40778	0	0	0	0	0.00066534
CptB	0	**0.93562**	0.25864	0	0	0	0.31488	0.00068129
CptC	0.39868	0.26109	**0.92148**	0	0	0	0	0.00029365
CptD	0	0	0	**0.93371**	0	0.25732	0.31475	0.00093133
CptE	0	0	0	0	**0.90912**	0.40579	0	0.00099834
CptF	0	0	0	0.25951	0.3988	**0.91462**	0	0.00046992
CptG	0	0.31596	0	0.31665	0	0	**0.91514**	0.00045409
resto	0.035169	0.023693	0.010085	0.023444	0.033661	0.010088	0.0086117	0.18239

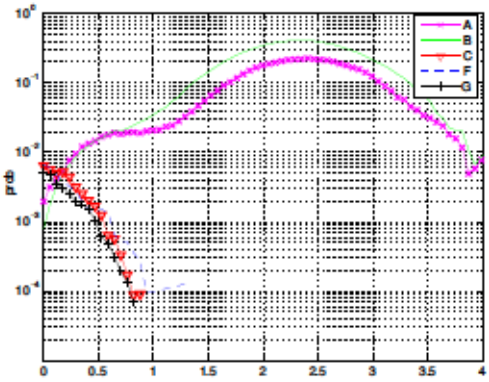
Tabla 4.14: Matriz de similitud de Jaccard para $\phi = 0$ para una matriz *in silico* generada siguiendo una distribución binomial negativa.

4.4.3.2 Co-agrupamientos basados en ϕ-conceptos con genes propios

Conviene estudiar en detalle cómo afecta ϕ a las probabilidades de definir los grupos correctamente. En las Figuras 4.19 aparecen las curvas de probabilidad donde se puede ver lo parecidas que son a las curvas de simulación de microarrays de las Figuras 4.11 y 4.15.



(a) Probabilidad asignar correctamente un gen al grupo dado.



(b) Probabilidad de asignar a un grupo un gen que no pertenece a ese grupo.

Figura 4.19: Prestaciones en función de ϕ para la matriz de expresión simulada con una función de probabilidad binomial negativa.

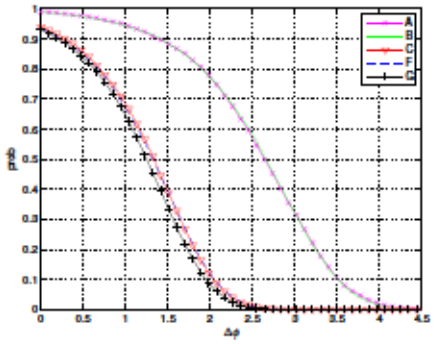
ϕ versus $\Delta\phi$ para co-agrupamientos basados en ϕ-conceptos con genes propios El valor de la curvas de probabilidad para $\Delta\phi$, donde recordemos que se define como el rango de valores de ϕ por el cual un gen pertenece exclusivamente a un grupo, se muestra en la Figura 4.20 y su forma recuerda mucho a las obtenidas para microarrays simulados en las Figuras 4.12 y 4.16.

Parece ser que las curvas de probabilidad son bastante parecidas con independencia del método que se utilice para simular la matriz de expresión genética, siempre y cuando los grupos sean parecidos como ha sido el caso en esta sección.

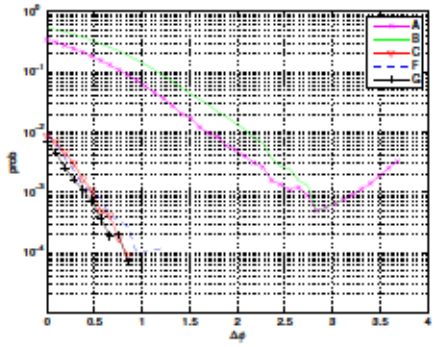
4.4.4 Conclusiones

En esta sección se han presentado varias matrices de expresión genética *in silico* creadas siguiendo diferentes algoritmos. Los grupos obtenidos mediante el análisis se han podido comparar con los originales y se ha visto que hay pocas diferencias en función del algoritmo elegido para generar las matrices.

Sólo se ha realizado el análisis en el dominio minplus para obtener los grupos de genes sobre-expresados, por lo tanto la exploración realizada ha estado en el rango $\phi \in [0, \infty)$. El análisis dual en el dominio maxplus para obtener los genes infra-expresados no se ha realizado porque el objetivo de esta sección es simplemente mostrar diferentes algoritmos para crear



(a) Probabilidad asignar correctamente un gen al grupo dado.



(b) Probabilidad de asignar a un grupo un gen que no pertenece a ese grupo.

Figura 4.20: Prestaciones en función de $\Delta\phi$ para la matriz de expresión simulada con una función de probabilidad binomial negativa.

matrices *in Silico* y ver la evolución de los retículos de conceptos a medida que varía el umbral ϕ. En este sentido añadir la exploración maxplus no habría aportado ningún elemento nuevo.

Observando los retículos de la Figura 4.9 se ve cómo a medida que disminuye el valor de ϕ los conceptos más cercanos al ínfimo ($\perp$) van acumulando objetos (genes) porque cada vez tienen más atributos. Así parece lógico que cuanto más cercano a 0 sea el valor de ϕ más conceptos aparezcan (Figura 4.8) y los conceptos formales que ya existían aumentan el número de objetos. Esto hace que la probabilidad de detectar correctamente un gen que pertenece a un grupo determinado aumente a expensas de aumentar también la probabilidad de asignar incorrectamente dicho gen al grupo equivocado, estas curvas de probabilidad se han mostrado en las Figuras 4.10, 4.14 y 4.18. Esto significa que para valores más altos de ϕ –o de forma dual más bajos de φ en maxplus– los genes que aparecen tienen una mayor probabilidad de pertenecer al co-agrupamiento seleccionado porque su *probabilidad de falso positivo* es más baja tal y como se adelantó en la Sección 3.3. A medida que ϕ disminuye, la probabilidad de falso positivo en cada uno de estos conceptos formales tiende al valor indicado en la Tabla 4.4.

Los conceptos formales generan un sistema de agrupamiento jerárquico en el cual unos conceptos están incluidos en otros, como se muestra en la Figura 4.1b. Si se desean obtener los genes que pertenecen de forma exclusiva a un concepto formal hay que recurrir a otra aproximación con otras curvas de probabilidad tal y como se explicó en las secciones 4.4.1.2, 4.4.2.2 y 4.4.3.2. En muchos casos esto será lo que habrá que realizar porque se desea conocer los genes expresados exclusivamente en unas muestras determinadas. En este caso es más difícil determinar el valor deseado de ϕ en la misma matriz para algunos grupos las curvas de probabilidad son monótonas decrecientes o decrecientes, pero en otros grupos como *A* y *B* la curvas de probabilidad presentan máximos y mínimos.

Para tratar de salvar el obstáculo del valor de umbral óptimo en el caso de grupos generados por los genes que pertenecen en exclusiva a un concepto formal se creó la idea de $\Delta\phi$, que indica el rango en el cual un gen pertenece de forma exclusiva en un concepto formal. Como se ha visto en las Figuras 4.12, 4.16 y 4.20 cuando mayor es el valor de $\Delta\phi$ generalmente

más baja es la *probabilidad de falso positivo*. Por este motivo se decidió crear el sistema de visualización explicado en 3.4.4 donde fácilmente se pueden ordenar los genes de un grupo en función de $\Delta\phi$.

4.5 Una discusión sobre el efecto de la normalización

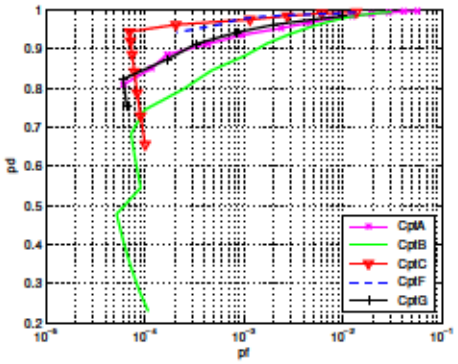
En la Sección 3.2 se explica el preprocesado que se puede aplicar a una matriz de expresión genética. Podemos distinguir dos tipos: uno, el que se puede aplicar teniendo algún conocimiento previo de la matriz de expresión (por ejemplo, como podría ser la relación entre varias filas que pertenecen a un mismo gen) y dos, la normalización de datos para preparar la matriz como entrada de análisis $\mathcal{K}$-FCA. En esta sección presentamos los resultados de los cuatro tipos de normalización propuestos:

- Media geométrica: Tal y como se describe en la fórmula 3.2.2. Es la normalización que se ha aplicado a todas las matrices de la Sección 4.2.
- Media aritmética según la ecuación 3.2.1.
- Filas con media 0 y varianza 1: como se explica en la Sección 3.2.2.2. Concretamente para los ejemplos se utilizará la fórmula 3.2.4.
- Normalización por filas y columnas con media 0 y varianza 1: añadiendo al anterior la normalización por columnas de la Sección 3.2.2.3.

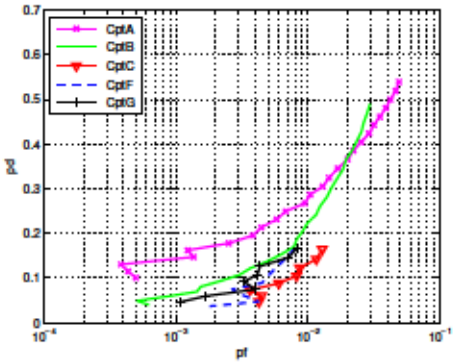
Partiendo de la matriz *in silico* generada para la Sección 4.3.1, se aplica cada una de las normalizaciones anteriormente descritas. A continuación se calcula la *probabilidad de detección* (pd) junto con la *probabilidad de falso positivo* (pf) para diferentes grupos en las diferentes normalizaciones. En la Figura 4.21 se muestra la relación entre ambas probabilidades para cada una de las normalizaciones.

Cuando mayor sea la probabilidad de detección y menor sea la de falso positivo mejor será la clasificación. En este caso, el mejor entre todos los métodos de preprocesado es el que utiliza la media geométrica (Figura 4.21a). La media aritmética funciona bastante mal al tratarse de medidas que siguen una distribución exponencial.

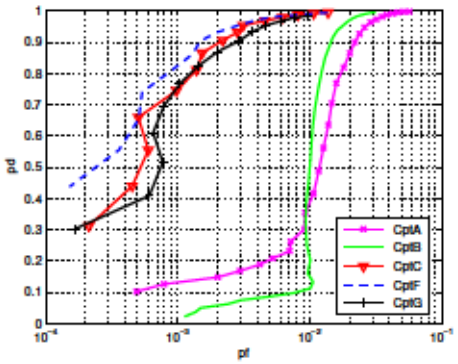
La normalización por filas para conseguir media 0 y varianza 1 (Figura 4.21c) no consigue el mismo nivel de probabilidad de falso positivo para una probabilidad de detección dada que la normalización por media geométrica. El problema al añadir además la normalización por columnas es que se obliga a guardar cierta relación a genes que no tienen nada que ver entre sí En la Figura 4.22 se muestra el nivel de expresión tras esta normalización, se puede ver como parte del grupo C desaparece y se confunde con el grupo A, con el cual se solapa en 2 columnas. Además las últimas filas que no pertenecen a ningún grupo tras esta normalización presentan unos niveles similares a los de los grupos conocidos. Esto hace que la probabilidad de falso positivo sea bastante mala para esta normalización en este ejemplo.



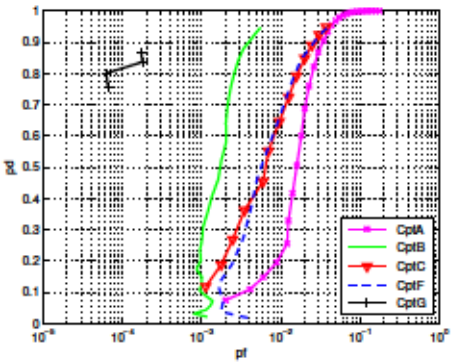
(a) Media geométrica



(b) Media aritmética



(c) Normalización por filas con media 0 y varianza 1



(d) Normalización por filas y columnas con media 0 y varianza 1

Figura 4.21: *Probabilidad de detección* para cada uno de los algoritmos de preprocesado. En el eje de abscisas se muestra la *probabilidad de falso positivo* y en el de ordenadas la probabilidad de detección. Los diferentes puntos de cada curva corresponden a diferentes valores de ϕ

4.6 Comparación con otros algoritmos

A pesar de que es difícil encontrar algoritmos que sean totalmente comparables teniendo en cuenta las características de $\mathcal{K}$-FCA (co-agrupamiento no partitivo y jerárquico enmarcado y apoyado por un procedimiento de EDA) y teniendo muy presentes las limitaciones de la generación de los datos in silico que hemos descrito, hemos explorado las prestaciones de los métodos que describimos en 2.1.2 utilizado diversas librerías públicas escritas en Matlab y R y proporcionadas, en la medida de lo posible, por los autores de cada uno de los algoritmos. También, salvo indicación contraria, hemos respetado las configuraciones que los autores indicaban, y que presumimos en cierto modo adaptadas a los experimentos que se describen en cada uno de los artículos. Esto puede ser el causante de que en varios casos nos encontremos con prestaciones bastante bajas sobre nuestras matrices in silico.

Así, y de nuevo salvo que especifiquemos lo contrario, la entrada de todos estos algoritmos es el logaritmo de la matriz de expresión genética M explicada en la Sección 4.2 y Figura 4.1a generada según el algoritmo de Ingrid & Speed, sin realizar la normalización por filas. Es

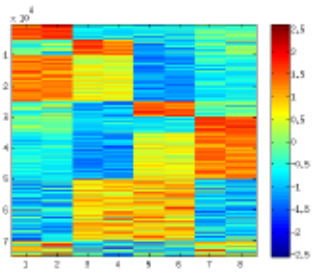


Figura 4.22: Matriz del nivel de expresión *in silico* normalizado por filas y columnas

decir, la entrada a todos estos algoritmos es $log(M)$ a no ser que se indique lo contrario. Se puede leer una breve introducción de cada algoritmo aquí utilizado en el capítulo 2.

4.6.1 K-means

Uno de los algoritmos de agrupamiento más populares es el k-means [20]. Este algoritmo se utiliza con bastante frecuencia para encontrar grupos en matrices de expresión genética [24, 26, 73].

El algoritmo k-means agrupa a todos los elementos en un grupo por filas (o columnas), no realiza co-agrupamientos, no permite que haya elementos sin clasificar, por lo que en este caso habrá que configurarlo para buscar $k = 8$ grupos que idealmente recogerán los 7 grupos que aparecen en la Figura 4.1a junto con todos los elementos que no pertenecen a ningún grupo. Como métrica hemos utilizado la distancia euclídea. Los resultados para la mayoría de los grupos son bastante buenos como muestra la matriz de confusión que aparece en la Tabla 4.15, en las columnas se agrupan los genes que pertenecen a los grupos conocidos y en las filas aparecen los genes de los grupos identificados por k-means.

	A	B	C	D	E	F	G	resto	pf
c1	**4999**	0	1	0	0	0	0	52	$\frac{53}{5052} = 0{,}0105$
c2	0	**4994**	0	0	0	0	2	70	$\frac{72}{5066} = 0{,}0142$
c3	0	0	**7536**	0	0	0	0	0	$\frac{0}{7536} = 0$
c4	1	2	**7463**	0	0	0	0	0	$\frac{3}{7466} = 4 \cdot 10^{-4}$
c5	0	0	0	**4955**	0	5	1	4820	$\frac{4826}{9781} = 0{,}493$
c6	0	0	0	0	**4999**	18	0	57	$\frac{75}{5074} = 0{,}0148$
c7	0	0	0	4	1	**14977**	1	0	$\frac{6}{14983} = 4 \cdot 10^{-4}$
c8	0	4	0	41	0	0	**19996**	1	$\frac{46}{20042} = 0{,}0023$
pd	$\frac{4999}{5000}$	$\frac{4994}{5000}$	$\frac{14999}{15000}$	$\frac{4995}{5000}$	$\frac{4999}{5000}$	$\frac{18}{15000}$	$\frac{1}{20000}$		

Tabla 4.15: Matriz de confusión para el algoritmo k-means con 8 grupos.

Pero la matriz $log(M)$ aunque se ha creado pensando en parecer lo máximo posible a una matriz de expresión genética real no lo es tanto ya que todos los genes de un grupo no tienen que compartir la misma media. Lo que interesa del agrupamiento de genes es la agrupación basada en cómo varían los genes en función de los distintos experimentos a los que los tejidos se ven sometidos. Por eso se realiza el preprocesado de la ecuación 4.4.1. Si

se utiliza la matriz *R* como variable de entrada y se selecciona como máximo 15 grupos la matriz de confusión que aparece se muestra en la Tabla 4.16.

	A	B	C	D	E	F	G	resto
c1	**4992**	0	1	0	0	0	0	4
c2	0	**4968**	4	0	0	0	25	3
c3	0	0	**5739**	0	0	0	0	0
c4	2	8	**4827**	0	0	0	0	0
c5	0	0	**4423**	0	0	0	0	0
c6	0	0	0	**4977**	0	2	20	6
c7	0	0	0	0	**4974**	0	0	1
c8	0	0	0	0	0	**3813**	0	0
c9	0	0	0	11	0	**3499**	0	0
c10	0	0	0	0	12	**3946**	1	0
c11	0	0	0	0	0	**3733**	0	0
c12	0	2	0	4	0	0	**10174**	0
c13	0	9	0	3	0	0	**9780**	1
c14	0	0	0	5	14	7	0	**2511**
c15	6	13	6	0	0	0	0	**2474**

Tabla 4.16: Matriz de confusión para el algoritmo k-means con 15 grupos.

Al ver la matriz de confusión se podría pensar que el algoritmo k-means ha sido capaz de identificar que el grupo C está formado por 3 subgrupos de media diferente como se explicaba en la Sección 4.3, pero si miramos el detalle de la partición no es así, la división del grupo C en tres grupos se debe a que la media de los tres subgrupos es tan similar en comparación con la varianza que no es diferenciable por k-means. Lo mismo que ha ocurrido con el grupo F que se encuentra dividido entre los grupos c8-c11 y el G c12-c13.

Estos ejemplos eran con una matriz de expresión que simulaba ser un microarray tal y como se explicaba en la Sección 4.3.1. Para comprobar si habría algún efecto a la hora de cambiar el algoritmo de generación del matriz de expresión, se decidió probar con la matriz generada en la Sección 4.3.3 donde se simulaba un tipo de secuenciación de próxima generación.

En este caso la matriz *M* que se obtenía de la simulación daba resultados muy pobres a la hora de identificar los grupos y por ello decidimos usar la matriz *R* como matriz de entrada al algoritmo k-means, es decir, la matriz de expresión normalizada con la ecuación 4.4.1. La matriz de confusión se muestra en la Tabla 4.17.

4.6.2 Cheng & Church

La implementación utilizada fue la escrita por [97] en Matlab. Con los parámetros por defecto el algoritmo encuentra 90 grupos, pero la mayoría son grupos con unas pocas decenas de genes, así que ensayamos distintos números de grupos hasta llegar a 12. En la Tabla 4.18 se muestra la matriz de confusión donde cada columna representa el grupo esperado y cada fila es el grupo clasificado por el algoritmo. En este caso el grupo C aparece divido en dos grupos c3 y c4, los grupos c9, c10, c11 y c12 no tienen un grupo claro definido, por eso aparecen en la categoría de *resto*.

	A	B	C	D	E	F	G	resto	pf
c1	**4994**	1	91	0	0	0	0	1737	$\frac{1829}{6823} = 0{,}268$
c2	0	**4994**	88	0	0	0	137	996	$\frac{1221}{6215} = 0{,}196$
c3	6	5	**14821**	0	0	0	0	3	$\frac{14}{14835} = 9{,}4 \cdot 10^{-4}$
c4	0	0	0	**4994**	0	98	167	1405	$\frac{1670}{6664} = 0{,}25$
c5	0	0	0	0	**4998**	120	0	840	$\frac{960}{5958} = 0{,}161$
c6	0	0	0	3	2	**14782**	0	4	$\frac{9}{14791} = 6{,}1 \cdot 10^{-4}$
c7	0	0	0	3	0	0	**12225**	15	$\frac{18}{12243} = 0{,}00147$
c8	0	0	0	0	0	0	7471	0	
	$\frac{4994}{5000}$	$\frac{4994}{5000}$	$\frac{14821}{15000}$	$\frac{4994}{5000}$	$\frac{4998}{5000}$	$\frac{14782}{15000}$	$\frac{12225}{20000}$		
pd	0,998	0,998	0,988	0,998	0,999	0,985	0,611		

Tabla 4.17: Matriz de confusión para el algoritmo k-means para una matriz de expresión simulada mediante la técnica explicada en la Sección 4.3.3.

Existen muchos otros genes que no son clasificados por el algoritmo en ninguno de los grupos. Si se aumentara el número de grupos que proporciona el algoritmo como salida se crearían nuevos grupos que contendrían los genes que no se han clasificado anteriormente, en cambio los genes que ya están clasificados no cambiarían. Por lo tanto la matriz de confusión sería igual pero con un mayor número de filas.

	A	B	C	D	E	F	G	resto	pf
c1	**4358**	0	0	0	0	0	0	7	$\frac{7}{4365} = 0{,}0016$
c2	0	**4271**	2	0	0	0	0	12	$\frac{14}{4285} = 0{,}0033$
c3	70	17	**8145**	0	0	4876	0	5	$\frac{4968}{13113} = 0{,}379$
c4	0	0	**2097**	0	0	722	0	0	$\frac{722}{2819} = 0{,}256$
c5	0	0	1	**4060**	432	934	1	62	$\frac{1430}{5490} = 0{,}26$
c6	0	0	0	0	**4150**	0	0	1	$\frac{1}{4151} = 2{,}4 \cdot 10^{-4}$
c7	0	0	3519	0	0	**7346**	0	0	$\frac{3519}{10865} = 0{,}323$
c8	0	378	0	325	0	0	**19709**	202	$\frac{905}{20614} = 0{,}044$
c9	0	0	0	0	361	7	0	**4031**	
c10	252	0	1	0	0	0	0	**409**	
c11	0	0	0	478	0	0	1	**172**	
c12	0	0	237	0	0	40	0	**0**	
	$\frac{4358}{5000}$	$\frac{4271}{5000}$	$\frac{10242}{15000}$	$\frac{4060}{5000}$	$\frac{4150}{5000}$	$\frac{7346}{15000}$	$\frac{19709}{20000}$		
pd	0,872	0,854	0,683	0,812	0,83	0,49	0,986		

Tabla 4.18: Matriz de confusión para el algoritmo de Cheng&Church

4.6.3 BiMax

Aunque la entrada del algoritmo origina utiliza matrices binarias, la implementación [97] realiza esa conversión. Este algoritmo realiza una clasificación jerárquica con grupos contenidos dentro de otros al igual que realiza FCA con sus conceptos formales. Por lo tanto no se comparará con los grupos descritos en la Tabla 4.1 sino con los conceptos formales de la Tabla 4.2. Un problema es que se generan cientos de grupos y es difícil decidir cuales son los de interés y cuáles no, este problema lo resuelve $\mathcal{K}$-FCA mediante la exploración y la representación de retículos como los vistos en la Figura 4.9. Aquí no se dispone de dicha ayuda

visual pero si se seleccionan a mano los grupos de interés entre los 255 que genera el algoritmo la matriz de confusión parece ofrecer buenos resultados en algunos casos: su probabilidad de detección llega casi a 1 pero a base de aumentar sustancialmente la probabilidad de falso positivo. En la Tabla 4.19 se muestra la matriz de confusión.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG
c3	**19999**	17609	15000	4110	2023	1511	2092
c24	15494	**39998**	14999	22049	2032	1534	20000
c30	15494	17608	**14999**	2316	215	171	2092
c10	2013	22025	1519	**39998**	15488	15000	19998
c17	2024	4009	1494	17455	**19998**	15000	1983
c22	176	2157	127	17455	15488	**15000**	1983
c30	1555	22025	1519	22047	1575	1534	**19998**

Tabla 4.19: Matriz de confusión para el algoritmo BiMax.

En la matriz de Jaccard se puede observar mejor el resultado (Tabla 4.20) cuando se compara con la Tabla 4.8 que se ha generado mediante $\mathcal{K}$-FCA.

	CptA	CptB	CptC	CptD	CptE	CptF	CptG
c3	**0.7792**	0	0.37458	0	0	0	0
c24	0	**0.91838**	0.29718	0	0	0	0.33183
c30	0.36887	0.25617	**0.81384**	0	0	0	0.019758
c10	0	0	0	**0.91964**	0	0.29585	0.33227
c17	0	0	0	0	**0.78319**	0.37807	0
c22	0	0	0	0.25645	0.37008	**0.82295**	0.019957
c30	0	0.31471	0.027981	0.31496	0	0.026628	0.8263

Tabla 4.20: Matriz de similitud de Jaccard para el algoritmo BiMax.

4.6.4 Order Preserving SubMatrices, OPSMs

Este algoritmo ofrece unos resultados bastante pobres. Se pueden identificar 7 grupos de genes diferentes, pero estos grupos guardan muy poca relación con los definidos en la Sección 4.2. Claramente este algoritmo no funciona bien para las matrices *in silico* generadas en este ejemplo. Hay que tener en cuenta la naturaleza de este algoritmo que trata de agrupar genes cuyo nivel de expresión conserve un orden a través de diferentes muestras. En este caso tenemos grupos que están formados por 2 y 4 muestras, lo que parece insuficiente para que este algoritmo pueda encontrar algún patrón, motivo por el que no presentamos aquí los resultados numéricos.

4.6.5 Plaid

Este algoritmo identifica 8 grupos pero en algunos casos es difícil establecer una relación entre alguno de esos grupos y los que venían definidos dificultando el cálculo de las probabilidades de detección y falsa alarma. Su matriz de confusión se muestra en la Tabla 4.21.

	A	B	C	D	E	F	G	resto
c1	**4925**	32	2095	467	41	1016	0	1099
c2	0	**4561**	1600	74	0	313	13890	9
c3	2455	2402	**14369**	4	784	146	0	751
c4	9	1	**12932**	0	0	0	0	114
c5	0	0	0	**3211**	0	13499	0	0
c6	95	2	0	0	**4459**	12046	0	118
c7	0	311	52	121	0	370	**13875**	1
c8	1	57	854	1501	0	1716	13746	17

Tabla 4.21: Matriz de confusión para el algoritmo Plaid.

4.6.6 Expectation Maximization

En [44] proponen una forma para calcular grupos cuando la matriz de expresión se obtiene mediante métodos de secuenciación de próxima generación. Esta técnica se basa en suponer que los niveles de expresión siguen una distribución binomial negativa como la explicada en la Sección 4.3.3. Para aplicar este algoritmo se utilizó la librería escrita en R que proporcionan los autores.

Como en otros caso aquí descritos se probó con el logaritmo de la matriz de expresión generada según el algoritmo de Ingrid & Speed. Es necesario definir el número de grupos que se van a buscar y como en el caso del k-means se utilizarán 8, es decir los 7 grupos definidos anteriormente junto con todos los elementos que no pertenecen a ningún grupo ya que este algoritmo siempre asigna un grupo a los genes. La matriz de confusión se muestra en la Tabla 4.22.

	A	B	C	D	E	F	G	resto
c1	**4970**	0	4	0	0	0	0	461
c2	0	**4578**	0	0	0	0	9788	707
c3	13	0	**5642**	0	0	0	0	1
c4	17	149	**5109**	0	0	0	0	1434
c5	0	273	**4245**	0	0	0	0	0
c6	0	0	0	**4866**	0	4	10211	1012
c7	0	0	0	0	**4991**	47	1	825
c8	0	0	0	134	9	**14949**	0	560

Tabla 4.22: Matriz de confusión con el algoritmo EM para una matriz de expresión simulada mediante el método de Ingrid & Speed.

Los resultados no son todo lo buenos que se cabría esperar porque este algoritmo está diseñado teniendo en cuenta que la matriz de entrada va a seguir una distribución binomial negativa. En el momento en el los niveles de expresión no sigan la distribución supuesta los resultados empeorarán. Esto se puede ver en la Tabla 4.23 donde se usó una matriz de expresión genética generada siguiendo una distribución binomial negativa como se definió en la Sección 4.3.3.

	A	B	C	D	E	F	G	resto	pf
c1	**4992**	0	30	0	0	0	0	1225	$\frac{1255}{6247} = 0{,}2$
c2	0	**4988**	24	0	0	0	26	947	$\frac{997}{5985} = 0{,}166$
c3	8	4	**14946**	0	0	0	0	249	$\frac{261}{15207} = 0{,}0172$
c4	0	0	0	**4977**	0	37	26	871	$\frac{934}{5911} = 0{,}158$
c5	0	0	0	0	**4999**	31	0	1100	$\frac{1131}{6130} = 0{,}184$
c6	0	0	0	7	1	**14932**	0	259	$\frac{267}{15199} = 0{,}0176$
c7	0	1	0	16	0	0	**11083**	288	$\frac{305}{11388} = 0{,}026$
c8	0	7	0	0	0	0	8865	61	
pd	$\frac{4992}{5000}$ 0,998	$\frac{4988}{5000}$ 0,997	$\frac{14946}{15000}$ 0,996	$\frac{4977}{5000}$ 0,995	$\frac{4999}{5000}$ 0,999	$\frac{14932}{15000}$ 0,995	$\frac{11083}{20000}$ 0,554		

Tabla 4.23: Matriz de confusión con el algoritmo EM para una matriz de expresión simulada mediante la técnica explicada en la Sección 4.3.3.

4.6.7 FABIA

De vuelta con los algoritmos de clasificación basados en modelos de factores como FABIA [23] parecen mostrar un resultado bastante malo en el momento en el que la matriz de entrada no sigue el modelo de generación asumido.

Así al utilizar como entrada la matriz de expresión genética de la Sección 4.3.1 creada con varias gaussianas, el resultado es bastante malo como muestra la matriz de confusión en la Tabla 4.24.

	A	B	C	D	E	F	G	resto
c1	808	0	53	0	921	6	0	2475
c4	5	3429	0	0	0	6	0	4
c2	11	0	1518	947	1236	0	963	0
c3	7	29	95	426	1115	3680	423	0
c5	9	5	441	3	0	0	5	1
c6	6	3	23	1	3	1	0	0
c7	10	123	1404	0	5	0	1	0
resto	9142	1386	1212	13623	1712	1299	13608	12520

Tabla 4.24: Matriz de confusión del algoritmo FABIA para una matriz de expresión simulada mediante gaussianas de media constante como explica la Sección 4.3.1.

En nuestra opinión, el pobre resultado de FABIA se debe a que el algoritmo está diseñado para funcionar con GED con más muestras. En el ejemplo aquí visto se tienen pocas muestras con grupos formados por pocas columnas. Grupos con columnas de 2 o 4 muestras no proporcionan la información suficiente al algoritmo para que pueda encontrar una tendencia significativa en los genes de tal forma que puedan ser agrupados, de una forma similar a lo que le ocurre al algoritmo OPSM explorado en la Sección 4.6.4. Hemos analizado esta situación cambiando la matriz de expresión genética y creando diferentes agrupaciones: FABIA es incapaz de detectar grupos con tan sólo 2 o 4 columnas. Para ejemplos con más muestras, del orden de 50 o mayores como se muestran en [23] el algoritmo FABIA presenta mejores resultados.

4.6.8 Conclusiones

De todos los algoritmos que se han comparado los que posiblemente salgan mejor parados son el k-means y el de Cheng & Church. El resto de los algoritmos no se adaptó tan bien a los datos de entrada. Uno de los posibles motivos de un resultado tan pobre puede ser debido a que la matriz GED no cumpliera los requisitos esperados por el algoritmo bien porque no seguía la distribución esperada como sucedía en el caso del algoritmo *Expectation Maximization* o bien porque las dimensiones de los grupos no tienen las dimensiones mínimas necesarias como ocurre con *FABIA*.

Hasta ahora hemos presentado varios algoritmos de agrupamiento con una única matriz de GED, de los que se ha mostrado su matriz de confusión o de Jaccard. Para dar una visión más amplia de la calidad de los diferentes algoritmos introducimos una comparativa usando diferentes matrices *in silico*con diferente relación señal a ruido. Para ello, hemos partido de la matriz GED explicada en la Sección 4.2 y Figura 4.1a pero con diferentes valores de μ –la varianza se deja igual a 1 en todos los casos–. Para medir el desempeño de cada algoritmo de forma resumida se utilizó el coeficiente de Jaccard[73] que da un valor entre 0 y 1 donde 1 implicaría clasificación perfecta. El resultado se puede ver en la Figura 4.23 donde se comparan diferentes algoritmos de co-agrupamiento junto con $\mathcal{K}$-FCA para dos umbrales $\phi = 0$ y $\phi = 1{,}5$:

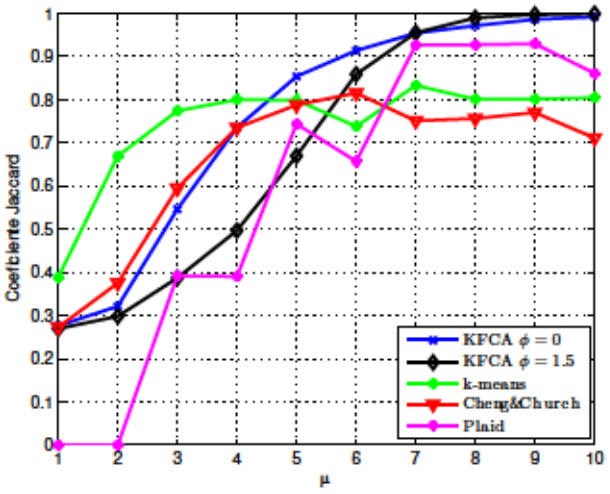


Figura 4.23: Coeficiente de Jaccard para matrices GED de la Sección 4.2 en función de μ y para distintos algoritmos de co-agrupamiento.

No se comparan todos los métodos de co-agrupamiento descritos en esta sección porque el resto poseen valores del coeficiente de Jaccard bastante bajos. En este caso el algoritmo k-means y el de Cheng & Church esperan encontrar 8 grupos, el algoritmo Plaid se deja con los valores por defecto de la implementación[97]. El algoritmo $\mathcal{K}$-FCA parece mostrar unos resultados bastante pobres cuando el valor de μ es bajo. Pero esto se debe a que aquí estamos realizando una clasificación automática con un umbral fijo para todos los grupos. Como se vio en la Sección 4.4, para el caso de co-agrupamientos basados en ϕ-conceptos con genes propios, cada grupo tiene un valor de umbral en el cual se maximiza la probabilidad de detección y otro donde se minimiza la probabilidad de falso positivo. Al suponer el mismo umbral se está

perdiendo capacidad de clasificación. Además estamos favoreciendo a los algoritmos k-means y el de Cheng & Church al proporcionar el número de grupos que esperamos encontrar.

Para tratar de generalizar estos resultados se mostrará a continuación la misma comparación pero con una matriz nueva. Esta matriz también se ha creado siguiendo el método de Ingrid & Speed, con 30000 filas y 8 columnas. Cada fila contiene el nivel de expresión de un gen para cada una de las 8 diferentes condiciones. En la matriz se definen 5 grupos, todos ellos tendrán unos niveles de expresión que seguirán una gaussiana de media μ y varianza 1:

- Grupo A: $[0, 4999] \times [0, 1]$
- Grupo B: $[5000, 9999] \times [6, 7]$
- Grupo C: $[10000, 14999] \times [0, 2]$
- Grupo D: $[15000, 19999] \times [3, 5]$
- Grupo E: $[20000, 24999] \times [1, 6]$

El resto de los elementos de la matriz que no se encuentran en ningún grupo siguen una distribución normal de media 0 y varianza 1. En la figura 4.24 se muestra un esquema de esta nueva matriz con sus grupos.

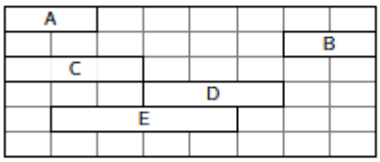


Figura 4.24: Matriz de expresión genética *in silico* creada siguiendo el método de Ingrid & Speed. Simula el resultado de 8 microarrays con tejidos sometidos a diferentes condiciones, cada microarray mide el nivel de expresión de 30000 genes.

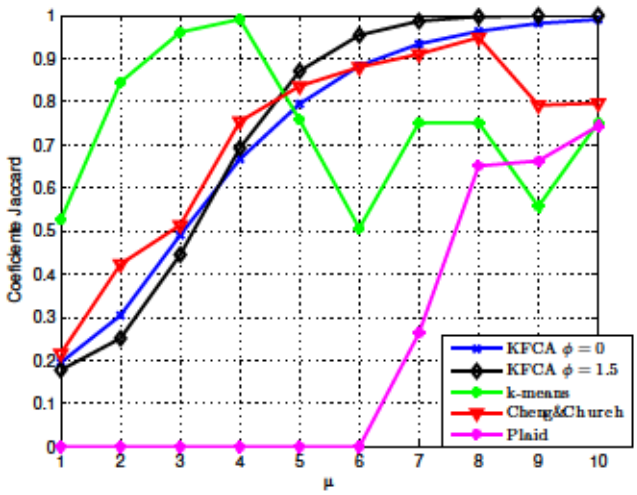


Figura 4.25: Coeficiente de Jaccard para matrices GED de la como la Figura 4.24 en función de μ y para distintos algoritmos de co-agrupamiento.

Los mismos algoritmos de coagrupamiento se utilizan para analizar esta matriz y se muestran en la Figura 4.25. Para k-means y Cheng & Church se indicó en 6 el número de grupos que se esperan encontrar. Los resultados parecen similares a los de las matrices antes analizadas.

4.7 Conclusiones

En este capítulo hemos abordado el complejo problema de evaluar los resultados del procedimiento exploratorio basado en FCA, objeto de esta tesis. Esta complejidad emana de la propia definición de problema del análisis de GED en donde los agrupamientos que se desea obtener no se conocen de antemano. En este capítulo hemos optado por la generación de matrices *in silico* para permitir la comprobación. En el Capítulo 5 lo complementamos con el análisis de matrices de expresión reales y recurriendo a recursos de información genética ya existentes para su comprobación, como explicaremos. Ambos modos de proceder tienen importantes limitaciones.

El principal inconveniente de la generación de matrices *in silico* proviene del sesgo que inevitablemente se introduce con la elección del método de síntesis de dichas matrices. No existe un método universalmente aceptado para la generación de dichos datos y aquí hemos utilizado tres distintos (dos para la simulación de datos procedentes de microarrays y uno para los secuenciadores de próxima generación). Hemos comprobado cómo muchos de los algoritmos utilizados para GED funcionan de una manera muy aceptable cuando las asunciones de generación de los datos coinciden con las de diseño del algoritmo, pero obtienen resultados muy malos cuando esas premisas no se cumplen.

Una de las ventajas importantes de nuestro marco de análisis de GED es que no hace ninguna suposición acerca de la distribución estadística de los datos de entrada. El soporte de EDA que hemos construido alrededor de la metodología $\mathcal{K}$-FCA proporciona un entorno de exploración visual que hace que, en los casos explorados, los resultados superen a los de los algoritmos de co-agrupamiento específicamente diseñados para GED. Por este motivo, abogamos por la adopción de esta metodología de exploración para GED. Así, la asistencia al investigador que realiza la exploración es el principal objetivo, renunciando a que el sistema resuelva de forma totalmente autónoma el problema puesto que, como hemos visto, depende altamente del cumplimiento de las asunciones de modelado que, hoy por hoy, no están lo suficientemente bien establecidas para esta disciplina.

No obstante, tenemos que señalar que hemos observado que algoritmos con un amplio espectro de aplicación como el k-means o el EM hacen una labor muy buena en muchos de los casos analizados aquí, en especial cuando el número de grupos que se pretende descubrir es conocido a priori. A pesar de que los experimentos que hemos realizados no son exhaustivos, se observa como los resultados no son tan buenos cuando dicho número no es conocido o cuando las distribuciones estadísticas de los datos no se ajustan a las asumidas en el caso del EM, estando de nuevo los resultados muy vinculados al método de generación de las matrices.

Un importante inconveniente de nuestro sistema, que impide su contraste adecuado con varios de los métodos de GED (FABIA, Plaid, OPSM y Bimax), es la limitación en el número de muestras o condiciones que es capaz de aceptar actualmente. Aunque en muchas ocasiones

es posible hacer una reducción de dicho número de muestras a base de preprocesado como explicamos en la Sección 3.2 esto es un trabajo que sería conveniente abordar en el futuro.

5

Análisis de expresión genética de datos reales

5.1 Introducción: recordatorio

Para ilustrar las potencialidades del $\mathcal{K}$-FCA presentamos aquí varios ejemplos de análisis entendidos como una guia a través de un proceso interactivo que se puede seguir para descubrir las relaciones entre genes. Todos estos análisis proporcionarán una idea de cómo deberían funcionar el conjunto de técnicas propuestas en esta tesis con datos reales de diferentes microarrays.

Estos análisis han sido realizados utilizando la herramienta web diseñada y construida para esta tesis llamada WebgeneKFCA[64], descrita brevemente en el Apéndice C.

A modo de recordatorio, recapitulamos brevemente el método de análisis justificado en los Capítulos 1 y 3:

1. El proceso aquí descrito enmarcado dentro de EDA comenzará con una fase de contextualización donde se define el problema y los datos de entrada. En estos casos serán diferentes ficheros CEL, cada uno representando una muestra, los que se combinarán para generar la matriz de expresión genética gracias al algoritmo RMA [66].

2. Una vez se tiene la matriz de expresión genética, se realizará el preprocesado (descrito en la Sección 3.2) y se prepararán los datos. En esta fase se puede incorporar conocimiento *a priori* que se tenga de la contextualización como puede ser el número de muestras repetidas utilizadas como control.

3. Después viene la fase de exploración, a partir de la cual la metáfora de LofK empieza a cobrar importancia. Se trabajará sobre los datos como si fueran un terreno sin cartografiar y mediante las piezas de información recopiladas en las diferentes fases de la exploración se irá completando dicho mapa

   (a) En una primera fase lo que se hace es calcular el número de conceptos en los diferentes retículos que aparecen para los diferentes umbrales posibles.

(b) A continuación, una vez vistos los puntos de interés se seleccionan los retículos más interesantes donde se ve que aparecen grupos significativos, y a partir de entonces sobre los retículos seleccionados se pueden realizar diferentes análisis buscando los genes de interés o midiendo la calidad de los grupos de interés.

4. Además se puede estudiar de forma individual cada gen viendo la evolución de los genes entre los diferentes grupos a medida que cambia el umbral o viendo cómo un grupo cambia en función del valor de umbral.

A continuación presentamos dos ejemplos de análisis.

5.2 Respuesta al Selenio en Arabidopsis thaliana

El selenio es un elemento químico que puede ser absorbido por las plantas utilizando las mismas rutas que utilizan para absorber azufre, dada su similitud química, pero un exceso de selenio se considera tóxico para las plantas y puede restringir su desarrollo. Para conseguir un mejor entendimiento de cómo afecta el selenio a las plantas se decidió probar con clones de la misma planta de *A. thaliana* en diferentes condiciones, en un suelo rico en selenio y en otro normal utilizado como muestra de control. El proceso completo al que se sometieron las plantas y su posterior análisis se puede consultar en [98].

Contextualización. Las muestras se dividieron en 8 grupos diferentes y cada una fue analizada en un microarray diferente, los microarrays usados eran del modelo *ATH1-121501*. Las muestras fueron las siguientes:

- **raiz1** y **raiz2**: Dos muestras extraídas de la raíz de las plantas de control.
- **raizSe1** y **raizSe2**: Dos muestras extraídas de la raíz de las plantas cultivadas en presencia de selenio.
- **brote1** y **brote2**: Dos muestras extraídas de los brotes de las plantas de control.
- **broteSe1** y **broteSe2**: Dos muestras extraídas de los brotes de las plantas cultivadas en presencia de selenio.

Esto genera una matriz GED de 22 810 filas, una por cada sonda, y 8 columnas. Todas las muestras se encuentran duplicadas para, de esta forma, mitigar los posibles efectos del ruido, así que realmente hay 4 casos diferentes que estudiar. Con estas muestras se podrá analizar los efectos que tiene el selenio en diferentes tejidos de una planta.

Preparación de los datos. Agrupamos el valor de diferentes sondas para mostrar el nivel de expresión de un gen directamente. El método utilizado fue el calcular el valor medio de todas las sondas que pertenecen a ese gen, tal y como se explica en la Sección 3.2.1.1. De esta forma se redujeron el número de filas de 22 810 a 21 324. Posteriormente a esta matriz se le aplicó una normalización donde el valor de expresión de cada gen se dividió por su media geométrica, tal y como se define en (3.2.2) donde r_{ij} es el nivel de expresión del gen i para la condición j.

Exploración: evolución del número de conceptos. Una vez normalizada la matriz, procedimos a a realizar la exploración $\mathcal{K}$-FCA para $\varphi \in (-\infty, 0]$ y $\phi \in [0, \infty)$, la evolución de cuyo número de conceptos se muestra en la Figura 5.1.

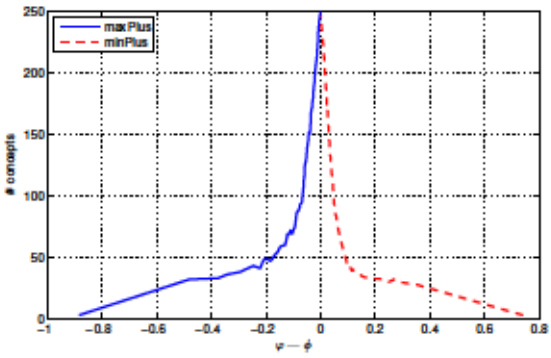


Figura 5.1: Número de conceptos en función del umbral φ y ϕ.

En esta gráfica parece que hay dos zonas de interés para la exploración: el intervalo $\varphi \in [-0{,}5, -0{,}2]$ para la infra-expresión, y el intervalo $\phi \in [0{,}1, 0{,}4]$ para la sobre-expresión.

Nótese que para valores de umbral cercanos a 0 el número de conceptos formales aumenta, la mayoría de ellos no representan agrupaciones con un significado biológico real sino que son producto del ruido intrínseco que existe en toda matriz GED.

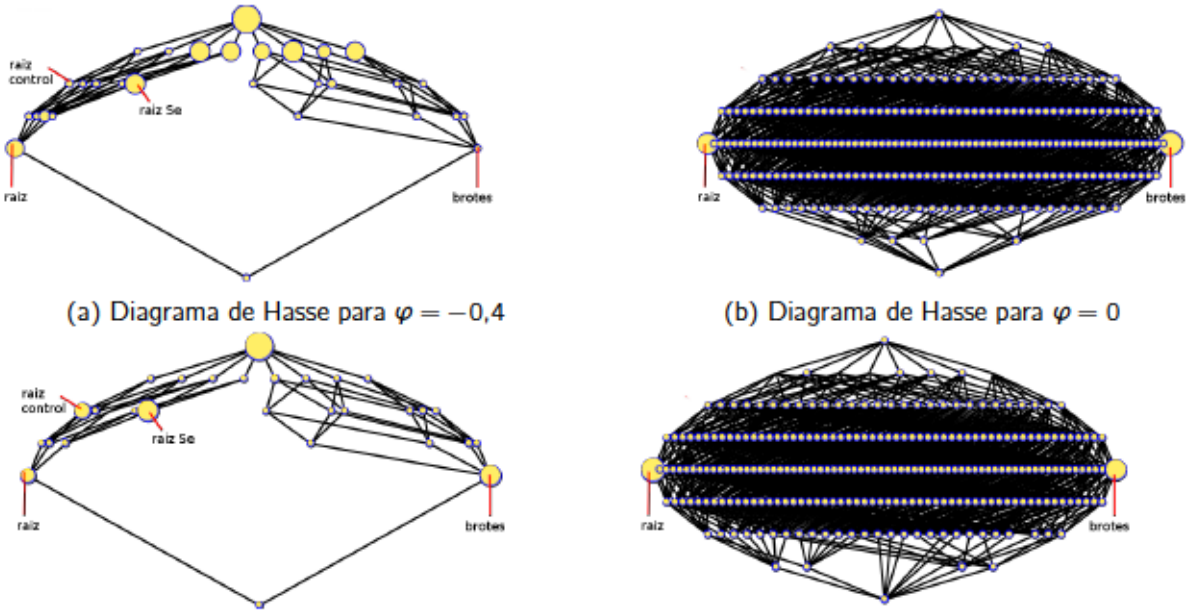
Exploración: co-agrupamiento de genes. En la Figura 5.2 se muestran los retículos de infra-expresión para los valores de $\varphi = 0$ y $\varphi = -0{,}4$ y los de sobre-expresión en $\phi = 0$ y $\phi = 0{,}33$. En los retículos de las Figuras 5.2b y 5.2d se ve claramente que hay dos grupos principales que corresponden con los genes sobre-expresados o infra-expresados para la raíz y los brotes, con independencia de que haya o no selenio en las muestras. Esto se debe a que hay genes que se encuentran muy diferenciados en cuanto nivel de expresión en los brotes y en la raíz, como cabría esperar.

Análisis de grupos de genes. Si por ejemplo se desea saber cual es el efecto del selenio en la expresión genética del tejido de la raíz sólo hace falta mirar al grupo correspondiente. En la Tabla 5.1 se muestran los genes sobre-expresados en la raíz bajo la presencia de selenio para $\phi = 0{,}2$. Se ve que, como respuesta al exceso de selenio, activa genes que responden a las ataques, heridas y situaciones de estrés que puede sufrir la planta. Esto es algo bastante consistente con lo que cabría esperar.

Es fácil ver que los umbrales iguales y nulos $\varphi = \phi = 0$ proporcionan, a lo sumo, informaciones muy genéricas y complementarias de la información de expresión, si la normalización es la correcta. En la Tabla 5.2 se muestran 30 términos GO del grupo que contiene los genes de la raíz infra-expresados en el dominio maxplus para $\varphi = 0$, este grupo sale de los genes que pertenecen en exclusiva al concepto formal dado por $\mu(\{raiz1, raiz2, raizSe1, raizSe2\})$. Evidentemente este es el grupo dual del que contiene los genes sobre-expresados en los brotes para $\phi = 0$ en el dominio minplus. Se ve como la mayor parte de los términos GO están

Término GO	Título	p-Value	genes en co-agrupamiento	genes en GO
GO:0010200	response to chitin	0.0000	39	122
GO:0009611	response to wounding	0.0000	24	145
GO:0006952	defense response	2.3744e-12	34	536
GO:0009873	ethylene mediated signaling pathway	2.7861e-11	18	155
GO:0006350	transcription	6.8610e-8	43	1162
GO:0009408	response to heat	1.6986e-7	13	136
GO:0045449	regulation of transcription	1.9791e-7	52	1609
GO:0009753	response to jasmonic acid stimulus	4.4468e-7	12	124
GO:0003700	sequence-specific DNA binding transcription factor activity	0.0000024847	47	1513
GO:0050832	defense response to fungus	0.0000058516	11	132
GO:0002679	respiratory burst involved in defense response	0.000013874	3	4
GO:0016563	transcription activator activity	0.000022220	10	125
GO:0010193	response to ozone	0.000026713	5	24
GO:0009644	response to high light intensity	0.000034330	6	41
GO:0006355	regulation of transcription, DNA-dependent	0.000046403	31	937
GO:0006979	response to oxidative stress	0.000047517	14	259
GO:0051865	protein autoubiquitination	0.00011732	3	7
GO:0009414	response to water deprivation	0.00012096	11	183
GO:0006950	response to stress	0.00019502	14	296
GO:0050691	regulation of defense response to virus by host	0.00023159	2	2
GO:0042542	response to hydrogen peroxide	0.00030009	5	39
GO:0009695	jasmonic acid biosynthetic process	0.00044231	4	24
GO:0005310	dicarboxylic acid transmembrane transporter activity	0.00068774	2	3
GO:0008271	secondary active sulfate transmembrane transporter activity	0.00069681	3	12
GO:0045941	positive regulation of transcription	0.0012571	5	53
GO:0015116	sulfate transmembrane transporter activity	0.0013931	3	15
GO:0009751	response to salicylic acid stimulus	0.0015726	7	111
GO:0008272	sulfate transport	0.0016953	3	16

Tabla 5.1: Términos GO que aparecen sobre-expresados en la raíz en presencia de selenio para $\phi = 0{,}2$. En la primera columna aparece el término GO al que se se refiere la fila, a continuación aparece el título asociado a dicho término. En la tercera columna aparece la probabilidad de que esos genes hayan acabado por azar en el mismo grupo. En la cuarta columna aparece el número de genes dentro del grupo que pertenecen a esa categoría y la última columna muestra el número de genes para ese término GO que puede detectar el microarray.



(a) Diagrama de Hasse para $\varphi = -0{,}4$

(b) Diagrama de Hasse para $\varphi = 0$

(c) Diagrama de Hasse para $\phi = 0{,}3$

(d) Diagrama de Hasse para $\phi = 0$

Figura 5.2: Retículos de sobre-expresión (fila de arriba) e infra-expresión (fila de abajo) usando $\mathcal{K}$-FCA para *A. thaliana*, para diferentes valores de φ y ϕ.

relacionados con la fotosíntesis o aparecen de alguna forma relacionados a los cloroplastos (orgánulos donde se realiza la fotosíntesis). Es bastante lógico que los genes relacionados con la fotosíntesis presenten un nivel de expresión mucho más bajo en la raíz que en los brotes.

El mismo ejercicio se puede realizar con el grupo que contiene los genes de los brotes infra-expresados para $\varphi = 0$, formado por los genes que pertenecen en exclusiva al concepto formal dado por $\mu(\{brote1, brote2, broteSe1, broteSe2\})$, que es el grupo dual del que contiene los genes sobre-expresados en la raíz para $\phi = 0$. En la Tabla 5.3 se muestra este grupo donde se ve que los términos GO están relacionados con el transporte de nutrientes y con la respiración de oxígeno, funciones que realiza la raíz con mayor intensidad que los brotes.

Como herramienta de análisis también podemos dar el retículo de genes y términos de GO como el mostrado en 5.3. En este retículo aparecen 21 genes están asociados con 44 términos GO.

Análisis de un gen en particular. Centrémonos en el gen identificado por Arabidopsis Genome Initiative (AGI) como AT1G17180 y también conocido por su identificador en NCBI como ATGSTU25. Según AGI y NCBI este gen codifica una proteína de la familia del *glutatión*, una enzima que se utiliza como respuesta a unos niveles altos de toxicidad. Esto encaja también con el termino GO:0009407 al que pertenece este gen.

Existe la posibilidad de ver la evolución de este gen a lo largo de los diferentes conceptos FCA: cómo, a medida que el valor de ϕ disminuye, el concepto al que este gen pertenece se va moviendo del concepto supremo ($\top$) al ínfimo ($\bot$). Este proceso se puede mostrar en un diagrama FCA donde aparecen todos los conceptos formales en los que puede estar el gen a medida que cambia su valor de ϕ, de una forma similar a como se explicó en la Sección 3.4.3.

Término GO	Título	p-Value	genes en co-agrupamiento	genes en GO
GO:0031977	thylakoid lumen	0.0000	48	60
GO:0009536	plastid	0.0000	523	842
GO:0009941	chloroplast envelope	0.0000	276	428
GO:0009543	chloroplast thylakoid lumen	1.1102e-16	66	78
GO:0009535	chloroplast thylakoid membrane	1.1102e-16	256	330
GO:0010287	plastoglobule	1.1102e-16	50	63
GO:0015979	photosynthesis	2.2204e-16	84	118
GO:0009507	chloroplast	3.3307e-16	1392	2851
GO:0009570	chloroplast stroma	3.3307e-16	305	433
GO:0009579	thylakoid	5.5511e-16	198	263
GO:0009658	chloroplast organization	9.9920e-16	37	47
GO:0015995	chlorophyll biosynthetic process	1.1846e-13	29	35
GO:0009295	nucleoid	3.5782e-13	27	32
GO:0009508	plastid chromosome	2.0946e-10	17	18
GO:0030095	chloroplast photosystem II	2.0946e-10	17	18
GO:0009773	photosynthetic electron transport in photosystem I	2.8006e-10	15	15
GO:0010319	stromule	4.7714e-10	27	38
GO:0009522	photosystem I	5.7985e-10	19	22
GO:0009793	embryo development ending in seed dormancy	1.2139e-9	124	328
GO:0009654	oxygen evolving complex	2.6061e-9	19	23
GO:0009523	photosystem II	2.6061e-9	19	23
GO:0009765	photosynthesis light harvesting	2.8058e-9	20	25
GO:0009534	chloroplast thylakoid	2.8638e-8	20	27
GO:0016168	chlorophyll binding	3.1864e-8	19	25
GO:0004176	ATP-dependent peptidase activity	9.2564e-8	19	26
GO:0010027	thylakoid membrane organization	1.1780e-7	16	20
GO:0008152	metabolic process	2.0575e-7	442	1553
GO:0005488	binding	2.1158e-7	411	1432
GO:0009707	chloroplast outer membrane	2.4439e-7	19	27
GO:0019252	starch biosynthetic process	3.4195e-7	13	15
GO:0019898	extrinsic to membrane	9.2855e-7	17	24

Tabla 5.2: Términos GO que aparecen infra-expresados en la raíz y sobre-expresados en los brotes para $\varphi = 0$. En la primera columna aparece el término GO al que se se refiere la fila, a continuación aparece el título asociado a dicho término. En la tercera columna aparece la probabilidad de que esos genes hayan acabado por azar en el mismo grupo. En la cuarta columna aparece el número de genes dentro del grupo que pertenecen a esa categoría y la última columna muestra el número de genes para ese término GO que puede detectar el microarray.

Término GO	Título	p-Value	genes en co-agrupamiento	genes en GO
GO:0016021	integral to membrane	0.0000	803	2342
GO:0005886	plasma membrane	0.0000	900	2184
GO:0006810	transport	0.0000	450	1223
GO:0005783	endoplasmic reticulum	0.0000	242	459
GO:0016020	membrane	9.9920e-16	1204	3550
GO:0005789	endoplasmic reticulum membrane	4.9960e-15	95	177
GO:0005773	vacuole	1.2934e-13	245	621
GO:0004601	peroxidase activity	2.8431e-12	67	119
GO:0015031	protein transport	6.3226e-12	152	356
GO:0009651	response to salt stress	1.4154e-11	157	374
GO:0005794	Golgi apparatus	2.7642e-11	145	341
GO:0006979	response to oxidative stress	2.9967e-10	114	259
GO:0005829	cytosol	5.3831e-9	184	485
GO:0042744	hydrogen peroxide catabolic process	6.4682e-9	47	84
GO:0016192	vesicle-mediated transport	8.8289e-9	80	173
GO:0005507	copper ion binding	6.5071e-8	104	250
GO:0000502	proteasome complex	7.4583e-8	35	59
GO:0010043	response to zinc ion	9.0964e-8	28	43
GO:0006886	intracellular protein transport	1.8415e-7	81	186
GO:0016491	oxidoreductase activity	2.8923e-7	393	1212
GO:0007264	small GTPase mediated signal transduction	4.6826e-7	48	96
GO:0055114	oxidation reduction	5.5625e-7	389	1205
GO:0045271	respiratory chain complex I	0.0000013275	28	47
GO:0016820	hydrolase activity, acting on acid anhydrides, catalyzing transmembrane movement of substances	0.0000019168	33	60
GO:0012505	endomembrane system	0.0000032966	851	2873
GO:0046686	response to cadmium ion	0.0000051373	136	372
GO:0004298	threonine-type endopeptidase activity	0.0000058412	17	24
GO:0005839	proteasome core complex	0.0000058412	17	24
GO:0004175	endopeptidase activity	0.0000065183	21	33
GO:0046274	lignin catabolic process	0.0000068717	14	18
GO:0016126	sterol biosynthetic process	0.0000079956	20	31

Tabla 5.3: Términos GO que aparecen infra-expresados en los brotes y sobre-expresados en la raíz para $\phi = 0$. En la primera columna aparece el término GO al que se se refiere la fila, a continuación aparece el título asociado a dicho término. En la tercera columna aparece la probabilidad de que esos genes hayan acabado por azar en el mismo grupo. En la cuarta columna aparece el número de genes dentro del grupo que pertenecen a esa categoría y la última columna muestra el número de genes para ese término GO que puede detectar el microarray.

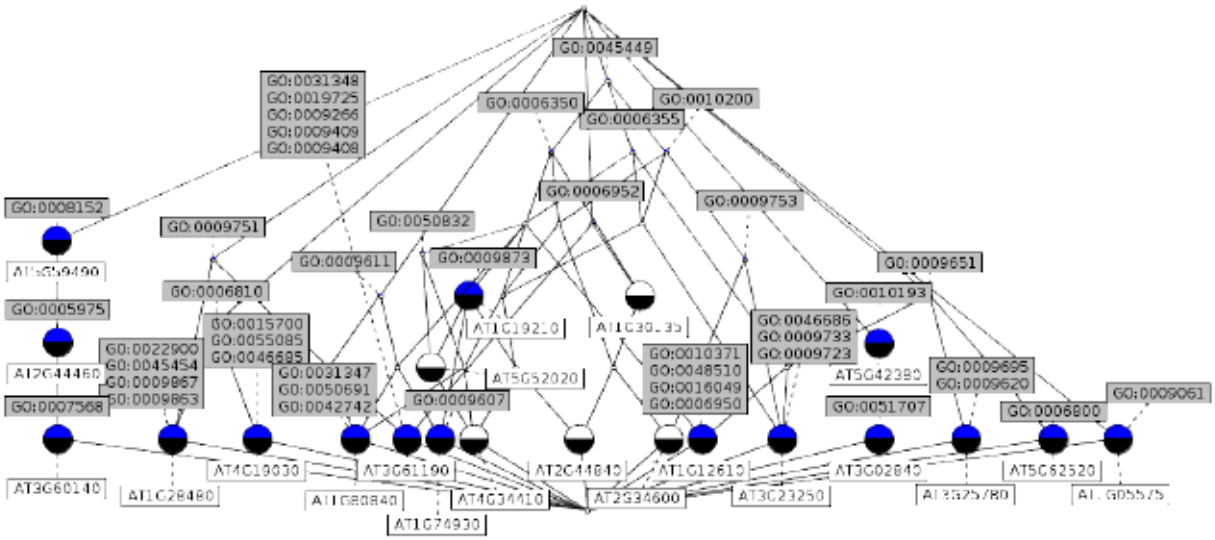


Figura 5.3: Relación entre genes y términos GO para los genes obtenidos para las muestras de la raíz con exceso de selenio y $\phi = 0{,}5$.

Se ve cómo este gen parece expresarse primero con genes de la raíz en presencia de selenio (raizSe1 y raizSe2) que con el resto de atributos. Así que es un gen que se encuentra expresado en la raíz de la planta mucho más que en los brotes y además parece tener una respuesta positiva a la intoxicación por selenio, algo que encaja con lo publicado en NCBI donde se encuentra información y artículos relacionados a la capacidad desintoxicadora que este gen presenta[1].

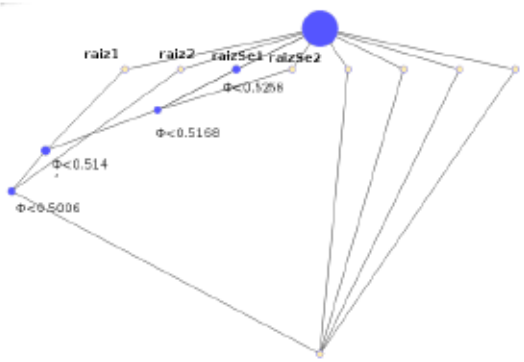


Figura 5.4: Evolución del gen AT1G17180 en función de ϕ.

En este punto, un artículo que usase este método seguiría con una sección de "findings".

5.3 Respuesta a Doxiciclina en células trisómicas con gen XIST

Esta sección se centrará en repetir en análisis de la matriz GED realizado por Jiang en [99]. Las hembras de los mamíferos poseen dos cromosomas X, mientras que los machos poseen

[1] http://www.ncbi.nlm.nih.gov/gene/?term=atgstu25

un cromosoma X y otro Y. Para evitar la sobre-expresión de genes del cromosoma X en las hembras el segundo cromosoma X se encuentra inactivo gracias al gen *XIST*. La idea que busca Jiang es insertar este gen en una copia del cromosoma 21 para poder inactivarlo en células humanas que sufran de trisomía en el par 21. Este gen *XIST* se insertó de tal forma que sólo se activará cuando se encuentre en presencia de *Doxiciclina*, así que se espera que ante la presencia de esta molécula una célula con trisomía del par 21 presente unos niveles de expresión similares a los de una célula disómica.

Contextualización. Se obtuvieron 27 muestras[2] de diferentes células analizadas con el microarray *PrimeView* de Affymetrix. Había 9 tejidos diferentes y tres réplicas para cada uno, los tejidos analizados fueron etiquetados como:

- **Male iPS**: Células pluripotenciales inducidas masculinas con *disomía* en el par 21. Se tienen tres muestras de este tipo y son utilizadas como control.
- **Parental no Dox**: Células pluripotenciales inducidas masculinas con *trisomía* en el par 21 y sin tratamiento. Se tienen tres muestras de este tipo y son utilizadas como control.
- **ParentalDox**: Células pluripotenciales inducidas masculinas con *trisomía* en el par 21 y tratada con *Doxiciclina*. Se tienen tres muestras de este tipo, son utilizadas como control para ver los efectos de la *Doxiciclina* en células pluritonenciales con *trisomía*.
- **Clone N no Dox**: Células pluripotenciales inducidas masculinas de células con *trisomía* en el par 21 y el gen *XIST* insertado. Aquí este gen no es expresado porque las muestras no se encuentran bajo el tratamiento de *Doxiciclina*. Se han realizado 3 clonaciones diferentes nombradas como: *Clone 1 no Dox*, *Clone 2 no Dox*, *Clone 3 no Dox*. Para cada una de estos clones se han tomado 3 muestras.
- **Clone N Dox**: Células pluripotenciales inducidas masculinas de células con *trisomía* en el par 21, con el gen *XIST* insertado y tratadas con *Doxiciclina*. El gen *XIST* debería expresarse en estas condiciones y producir la condensación de uno de los tres cromosomas 21. Se han realizado 3 clonaciones diferentes nombradas como: *Clone 1 Dox*, *Clone 2 Dox*, *Clone 3 Dox*. Para cada una de estos clones se han tomado 3 muestras.

Sólo se han seleccionado las sondas del microarray que están asociadas a un gen del cromosoma 21, así que la matriz de expresión a analizar tiene unas dimensiones de 621×27.

Preparación de los datos. Como se tienen 27 muestras diferentes y el retículo dibujado para una cantidad tan grande de conceptos (posiblemente 2^{27}) no producirá un resultado fácilmente inteligible, la solución por la que se opta en este caso es la de agrupar varios casos en uno, reduciendo así el número de columnas de la matriz, tal y como se explica en la Sección 3.2.1.2. Las columnas se agrupan de tres en tres, quedándose con la media geométrica de las muestras que corresponden a los mismos tejidos. De esta forma la matriz resultante tiene unas dimensiones de 621×9. A esta matriz se le aplica una normalización donde el valor de expresión de cada sonda se divide por su media geométrica, tal y como se define (3.2.2) donde r_{ij} es el nivel de expresión de la sonda i para la condición j.

[2] http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE47014

Exploración: evolución del número de conceptos. La evolución del número de conceptos se muestra en la Figura 5.5. Cambios repentinos en la pendiente revelan valores de existencia

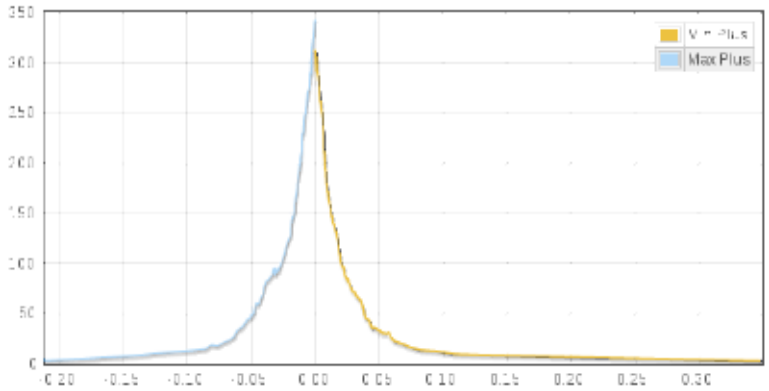


Figura 5.5: Número de conceptos en función del umbral φ y ϕ.

donde el retículo cambia sustancialmente y por tanto serán los valores de interés para mirar. En esta ilustración, los valores integrantes son $\varphi \in [-0{,}07, -0{,}02]$ y $\phi \in [0{,}02, 0{,}08]$.

Exploración: análisis de la infra-expresión. El orden de aparición de un gen en un grupo dado puede ser relacionado con la confianza de que ese gen pertenezca a dicho grupo. Si se selecciona un valor φ muy restrictivo, por ejemplo si $|\varphi| \gg 0$ (ver Figura 5.6 , $\varphi = -0{,}15$) entonces sólo los grupos que salen del retículo en la segunda fila (aquellos que están altamente infra-expresados en tres tipos de muestras, eso es, agrupaciones muy específicas), aparecen y el número de sondas que contienen es muy bajo. La mayor parte de las sondas pertenecen en exclusiva al concepto supremo, lo que significa que no podemos asegurar qué genes están infra-expresados con este umbral tan restrictivo. Cabe recordar que los co-agrupamientos específicos son aquellos cuya extensión (conjunto de sondas) sólo pueden ser declarados de una sola (si es de la segunda fila) o un par (si son de la tercera fila) de expresiones o muestras.

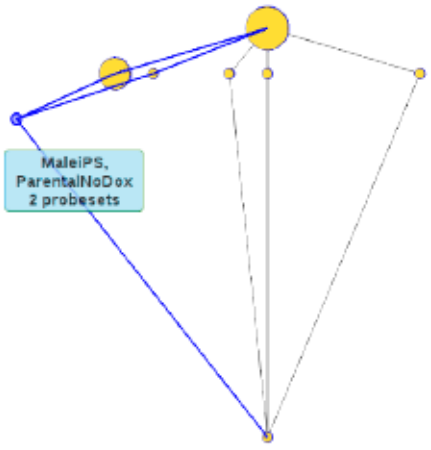


Figura 5.6: Retículo de infraexpresión genética para $\varphi = -0{,}15$.

A medida que se relajan las restricciones de φ más agrupaciones interesantes aparecen.

En particular, alrededor de $\varphi \simeq -0{,}05$ (Figura 5.7) grandes co-agrupamientos aparecen en la tercera, cuarta e incluso quita fila, lo que nos permita observar que combinaciones de muestras comparten el mismo número de genes infraexpresados.

Especialmente relevante es el co-agrupamiento correspondiente a con *Male iPS* y todos los *Clones Dox* (quinta fila) que aparecen en este nivel pero persiste (tiene un amplio rango como definimos en la Sección 3.4.4) hasta $\varphi = 0$. Si se observa este grupo en $\varphi = -0{,}005$, por ejemplo, aparecen 31 genes (que se corresponden con 57 sondas): ADARB1, AGPAT3, ATP5O, C21orf33, C21orf45, C21orf59, C21orf88, C21orf91, CCT8, COL6A1, CSTB, CXADR, DONSON, DSCR3, FAM165B, GART, GCFC1, HSPA13, IFNAR1, ITSN1, KRTAP11-1, MORC3, PDXK, PFKL, PRDM15, PTTG1IP, SLC37A1, SUMO3, TIAM1, TMEM50B, USP25. Esto significa que estos genes de los tejidos clonados tratados con Doxiciclina tienen un nivel de expresión genética similar al que presenta un tejido con células disómicas.

Tales resultados son consistentes con aquellos dados en el artículo original [99] donde sólo algunos genes de las células trisómicas se espera que bajen su nivel de expresión cuando el tercer cromosoma es silenciado.

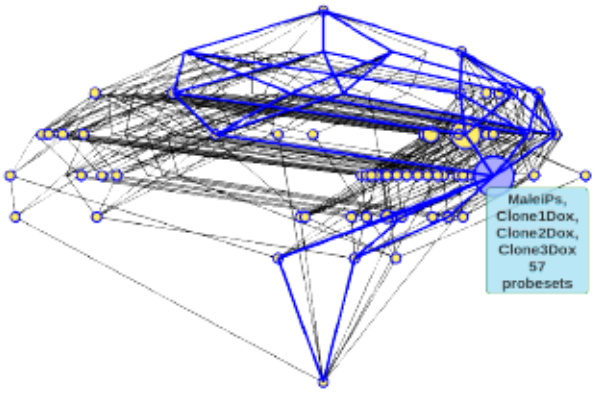


Figura 5.7: Retículo de infraexpresión genética para $\varphi = -0{,}005$.

Pero otros valores de φ pueden dar información adicional acerca cómo algunos genes son expresados y sobre otros posibles co-agrupamientos. Por ejemplo para $\varphi = -0{,}01$ (Figura 5.8) el mayor grupo con 43 sondas está formado por *Male iPS* y los *Clones* 1 y 3 tratados con Doxiciclina. Esto significa que por alguna razón *Clone 1* y *Clone 3* están respondiendo ligeramente mejor que *Clone 2* al experimento, lo cual es algo consistente con el gráfico encontrado en [99, Figura 4a]. Los genes correspondientes a estas sondas son: APP, GART, HSPA13, COL6A1, C21orf45, HMGN1, ITSN1, PSMG1, ATP5O, PCNT, HUNK, UBE2G2, ATP5O, C21orf33, GCFC1, CSTB, WRB, FAM165B, MRPL39, KRTAP19-1, C21orf59, CCT8, CRYZL1, SUMO3.

Exploración: análisis de la sobre-expresión. Como en los casos anteriores valores muy cercanos a $\phi = 0$ serán ruidosos y deberían ser evitados. El mayor grupo para $\phi = 0{,}004$ (Figura 5.9) es el formado por las condiciones *Parental no Dox*, *Parental Dox*, *Clone 1 no Dox*, *Clone 2 no Dox* y *Clone 3 no Dox* que son los tejidos que no están bajo tratamiento. Este grupo está cercanamente relacionado con el encontrado en $\varphi = -0{,}005$ para células disómicas

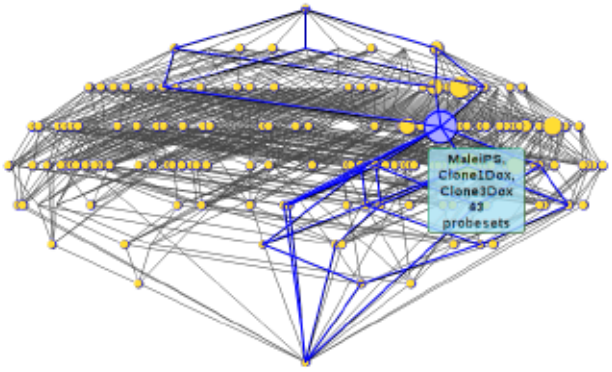


Figura 5.8: Retículo de infra-expresión genética para $\varphi = -0{,}01$.

y bajo tratamiento porque la infra/sobre-expresión es medida relativa a las condiciones duales. El grupo tiene 28 genes (41 sondas): AGPAT3, APP, ATP5J, ATP5O, C21orf45, C21orf57, C21orf88, COL6A1, CSTB, CXADR, DONSON, DSCR3, FAM165B, GART, HMGN1, HSPA13, IFNAR1, ITSN1, MORC3, PFKL, PTTG1IP, SLC37A1, SUMO3, TIAM1, TMEM50B, UBE2G2, USP25, ZNF295.

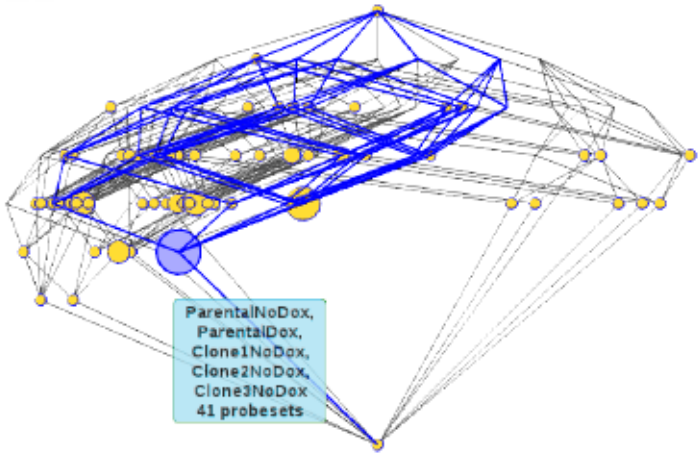


Figura 5.9: Retículo de sobre-expresión genética para $\phi = 0{,}004$.

Para diferentes valores de ϕ otros grupos emergen. Por ejemplo para $\phi = 0{,}05$, en la Figura 5.10, hay un gran grupo para el conjunto de condiciones *Clone 1 no Dox*, *Clone 2 no Dox* lo cual significa que esos tejidos comparten algún tipo de inesperado comportamiento.

Análisis de grupos de genes. Se puede buscar, por ejemplo, cómo son los genes infra-expresados que pertenecen al grupo dado por células que se espera que se comporten como células disómicas, esto es el dado por *Male iPS* y los *Clones Dox*. Los genes que aparecen en valores más bajos de φ, los que aparecen durante un mayor rango, son los que pertenecen con mayor probabilidad al grupo seleccionado.

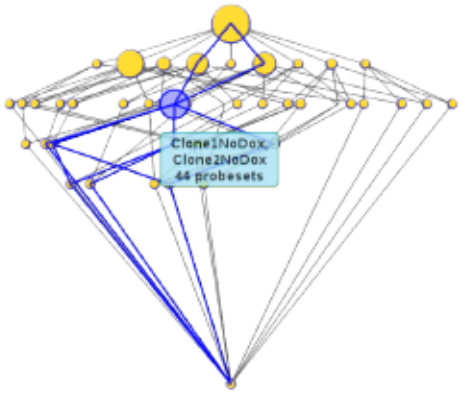


Figura 5.10: Retículo de sobre-expresión genética para $\phi = 0{,}05$.

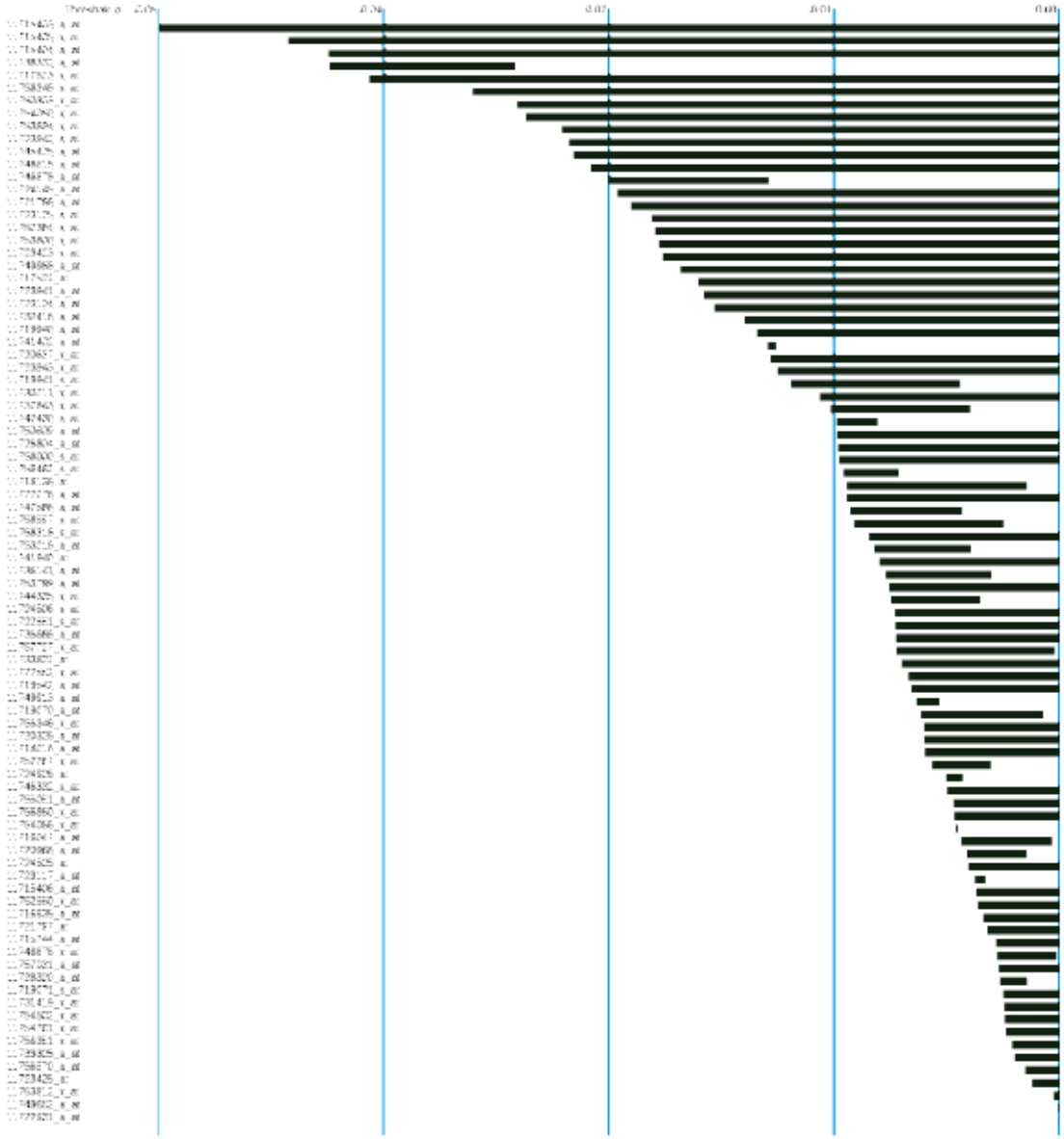


Figura 5.11: Rangos de aparición de las sondas infra-expresadas para los casos *Male iPS* y los *Clones Dox*.

6

Resumen de contribuciones, conclusiones y líneas futuras

6.1 Resumen de contribuciones

El objetivo de esta tesis no ha sido simplemente proporcionar un algoritmo cerrado para resolver un problema de co-agrupamiento específico. En esta tesis nos propusimos utilizar y expandir las capacidades de $\mathcal{K}$-FCA como un mecanismo de análisis de datos. En particular, para extender la metáfora de Wille sobre los retículos conceptuales y el FCA como "Paisajes del conocimiento", una forma de Análisis Exploratorio de Datos.

Centrándonos en los Datos de Expresión Genética (GED) como dominio de aplicación, los retículos conceptuales que surgen del $\mathcal{K}$-FCA, una versión de FCA adecuada para datos multivalorados, primero se postularon, y luego se estudiaron, como herramientas para co-agrupar genes en función del nivel de expresión mostrado para cada condición o experimento.

Como resultado de este estudio sintetizamos una metodología para el análisis exploratorio de expresión que demostramos con datos sintéticos, usados para comparar con otras aproximaciones al co-agrupamiento, y con datos reales: se realizó un primer análisis utilizando como ejemplo muestras reales para *A. thaliana* de una forma parecida a como se describe en la Sección 5.2, y también en el análisis de un racimo de genes en *Homo sapiens*.

Numerosos han sido los problemas que hemos tenido que resolver para llevar este empeño a buen puerto: para mejorar la visualización y facilitar el análisis se decidió desarrollar un nuevo software adaptado a las necesidades de visualización expuestas en la Sección 3.4, donde se pudieran comprobar los retículos y ver cómo cambian los conceptos formales con diferentes valores de φ/ϕ.

Los programas como Conexp[10] se centran en mostrar representación de retículos individuales pero no se garantiza la posición de un concepto formal de un experimento a otro. Pero para el caso de $\mathcal{K}$-FCA este tipo de representación acarrea problemas porque es difícil seguir el comportamiento de los conceptos a medida que cambia el umbral y aparecen nuevos conceptos formales en el diagrama. Por eso se diseñó e implementó el algoritmo descrito en la Sección 3.4.2 que permite comprobar retículos entre diferentes valores de umbral[64].

Con el objetivo de encontrar un marco al recién diseñado algoritmo se decidió crear la web WebgeneKFCA, explicada con detalle en el Apéndice C y de la cual se han sacado las

figuras de la Sección 5.1.

La web está orientada a ir descubriendo la información que guarda la matriz de expresión genética siguiendo el proceso exploratorio explicado en 3.1. Además se conecta con sistemas y bases de datos externas que permiten completar más la información recopilada, dar una medida de confianza, y transformarla en conocimiento como se explica en las Secciones 3.6 y 5.1.

Para concluir, a continuación listamos las principales contribuciones de la tesis:

1. Como resultado principal de esta tesis, se ha desarrollado una metodología y herramientas basadas en $\mathcal{K}$-FCA para el análisis exploratorio de datos de expresión genética.

   (a) La metodología se ha presentado en la Sección 3.8.1, y se ha ejemplificado sobre datos sintéticos *in silico* en el Capítulo 4 y sobre datos reales en el Capítulo 5.

   (b) La herramienta teórica fundamental, los análisis $(\overline{\mathbb{R}}_{\text{máx},+})$-FCA y $(\overline{\mathbb{R}}_{\text{mín},+})$-FCA adaptados al análisis de la infra- y la sobre-expresión, respectivamente, se han descrito en las Secciones 3.2 y 3.3.

   (c) La herramienta de visualización desarrollada para esta tesis, un nuevo tipo de diagrama para representar los retículos conceptuales basados en el mismo conjunto de atributos sin que cambie la posición de las posibles intensiones, se ha descrito en la Sección 3.4.2.

   (d) Se han desarrollado dos diagramas de Análisis Exploratorio de datos más para el contexto particular del análisis de datos genéticos, expuestos en las Secciones 3.4.3 y 3.4.4.

   (e) También se ha desarrollado un mecanismo de indexación de términos en ontologías externas mediante los grupos obtenidos en el análisis de datos para el Análisis de Enriquecimiento de Datos Genéticos en la Sección 3.7.

   (f) Finalmente, todo ellos se ha encarnado en un aplicación web de libre acceso, WebGeneKFCA, descrita en el Apéndice C.

   Estos resultados de investigación han quedado plasmados en dos artículos aceptados en congreso internacional especializados en FCA [17, 65], una herramienta en línea para favorecer el acceso Open Science a estos resultados de investigación [64] y en un artículo de revista enviado a una publicación especializada en análisis genético [100].

2. Como resultado secundario, hemos contribuido al desarrollo de la metáfora basada en FCA "Landscapes of Knowledge" como una metodología de exploración de datos.

   En particular hemos avanzado en las posibilidades de visualización que ofrece $\mathcal{K}$-FCA, mezclando diferentes retículos conceptuales en el mismo marco—reticulos de genes-condiciones, de genes-términos GO y retículos de evolución de un gen con el umbral—y en su uso como una técnica de indexación que permite introducir en el análisis exploratorio información de otras fuentes, en particular en el contexto del Análisis de Enriquecimiento Genético (GEA).

   Como ya hemos mencionado, este refinamiento del marco conceptual de los "Paisajes del Conocimiento" aplicado a la exploración de datos genómicos resultado de esta tesis ha sido objeto de publicación en revista internacional de alto factor de impacto [17].

6.2 Conclusiones

La tesis aquí expuesta propone que la metáfora de Wille acerca de los paisajes de conocimiento contenida en FCA también parece englobar las metáforas explicadas por las propuestas de Tukey para EDA. La formalización de datos inherente en los conceptos formales y su transformación en el concepto de retículo proporciona la bases para aplicar los paisajes de conocimiento en datos científicos como una especie de análisis exploratorio tal y como se adelantó en las Sección 1.5.

Esta metáfora se alinea bastante bien con el análisis exploratorio de datos en el sentido en el que la visualización y manipulación de un retículo de conceptos soporta un amplio rango de actividades relacionadas con la búsqueda científica asociada con la adquisición de conocimiento. Así en el Capítulo 3 se ha hecho hincapié en esta forma de análisis aplicado a matrices de expresión genética.

Los análisis aquí realizados muestran que esta herramienta interactiva de exploración de genes co-regulados produce unos resultados prometedores que ayudan al investigador a agrupar genes similares: en el contexto $\mathbb{R}_{\text{máx},+}$ captura los infra-expresados mientras que con $\mathbb{R}_{\text{mín},+}$ captura los sobre-expresados. En los Capítulos 4 y 5 se han mostrado estos análisis y se han comprobado con otros algoritmos de co-agrupamiento. De esta comparación la clasificación realizada mediante el análisis $\mathcal{K}$-FCA no ha salido nada mal parada respecto a algoritmos más complejos que hacen fuerte asunciones acerca de la distribución de datos esperada.

Además de todo esto se ofrece la oportunidad de mejorar el conocimiento de los grupos de genes utilizando bases de datos externas, enriqueciendo así la información obtenida como se explica en la sección 3.7. De la herramienta de análisis utilizada aquí, WebgeneKFCA, se ha publicado el código fuente en github[1]. Una descripción más detallada de esta herramienta se puede encontrar en el Apéndice C.

6.3 Líneas futuras

Gracias al gran avance que se está produciendo en el campo de la genética y a la reducción de precios a la hora de obtener matrices de expresión genética los estudios cada vez cuentan con más y más muestras con las que realizar los estudios. El proceso de análisis aquí descrito tiene una limitación a la hora de mostrar el resultado de matrices de expresión genética con muchos experimentos ya que el retículo de conceptos dibujado empieza a tener demasiados nodos. Hay que recordar que el máximo número de conceptos que puede aparecer es de 2^p donde p es el número de columnas de la matriz de expresión genética. Actualmente se proponen soluciones para reducir el número de columnas combinándolas como se explica en la Sección 3.2.1.2. Pero mediante este tipo de combinaciones se pierde información que podría ser de utilidad. Sería deseable disponer de un sistema de visualización que combinara conceptos similares y que de esta forma se redujera el número de nodos a representar.

[1] https://github.com/calabozo/webgenekfca

A

Introducción a la expresión genética

A.1 Introducción

En este apéndice se intenta proporcionar una visión general de diferentes procesos biológicos relacionados con la genética necesarios para entender el ámbito completo de esta tesis.

En la primera sección se hace una introducción al ADN, qué es y cual es su misión en la célula. A continuación en la sección A.3 se explica qué es ARNm y cual es su relación con la síntesis de proteínas descrita en A.4.

En la última sección se hace una introducción a la transcriptómica, es aquí donde se explican las diferentes formas que hay de medir el nivel de expresión de un gen en una célula.

A.2 ADN

El ADN es donde se encuentra la información genética de todos los seres vivos. La mayor parte de las células tienen ADN[1] que se divide y copia durante las fases de mitosis o meiosis.

La estructura del ADN es igual en todas las células, está formado por dos cadenas de nucleótidos que se enrollan sobre sí mismos formando la conocida estructura de doble hélice descubierta por James D. Watson y Francis Crick. El ladrillo básico de la doble cadena de ADN es el nucleótido que está formado por un un grupo fosfato (ácido fosfórico), desoxiribosa (monosacárido derivado de la ribosa al que le falta un átomo de oxígeno) y una base nitrogenada. En el ADN existen 4 tipos de bases nitrogenadas y son estas las que codifican toda la información almacenada en el ADN, son: adenina (A), timina (T), citosina (C) y guanina (G).

Los nucleótidos se unen entre sí formando una larga cadena mediante los grupos fosfatos. A su vez dos cadenas de nucleótidos de unen entre sí mediante los puentes de hidrógeno que se producen entre sus bases nitrogenadas complementarias en un proceso llamado *hibridación*. Así pues una base de adenina solo se unirá con una base de timina y una de citosina con una de guanina. Esto proporciona una especie de copia de seguridad y mayor estabilidad a la cadena de ADN. De esta forma la cantidad de adenina es igual a la de timina y la de citosina

[1] Existen células que durante su desarrollo se deshacen de todo el ADN de su núcleo como los glóbulos rojos

igual a la de guanina. Esta relación fue descubierta por Erwin Chargaff y desde su publicación se conoce como la regla de Chargaff [101]. Las terminaciones de las cadenas de ADN son designadas por 5' y 3', donde la terminación 5' tiene el grupo fosfato y el 3' el grupo OH. Las secuencias de ADN (o ARN) se enumeran desde la terminación 5' al 3'. En la figura A.1 se puede ver con detalle la estructura del ADN que se ha comentado.

En biología las células se pueden dividir en dos grandes grupos: *procariotas* y *eucariotas*.

Las células *procariotas* tienen todo el ADN en una cadena doble formando por un anillo [2]. Éstas son las células más simples, su principal característica es que carecen de núcleo.

Las células *eucariotas* tienen la mayor parte de su ADN protegido dentro del núcleo, donde se encuentra dividido en varias cadenas dobles llamadas cromosomas. Además en las eucariotas encontramos una cadena de ADN en forma de anillo, similar al que se encuentra en las procariotas, dentro de las mitocondrias [3] y de los cloroplastos.

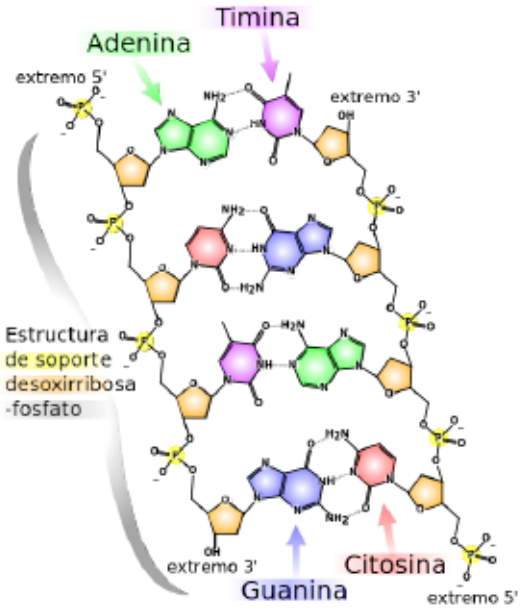


Figura A.1: Estructura del ADN. Fuente: Wikipedia

A.3 ARN

El ARN es una molécula muy parecida al ADN. Está formada for una cadena de ácidos nucleicos como el ADN pero presenta ligeras diferencias. En lugar de una molécula de desoxiribosa tiene una molécula de ribosa. En las bases nitrogenadas se sustituye la timina por el uracilo (U) que como esta es la base complementaria de la adenina. Además, a diferencia del ADN que aparece como una doble cadena con las bases nitrogenadas en su interior, el ARN generalmente está formado con una cadena simple de ácidos nucleicos. Esta cadena se pliega sobre si misma gracias a la atracción que hay entre sus bases complementarias (A-U y C-G) dando lugar a estructuras tridimensionales más o menos complejas. Existen varios tipos de ARN en función de la misión que desempeñan[102]:

ARNm El ARN mensajero es una molécula que codifica las instrucciones necesarias para crear una proteína. Es una copia de parte del ADN que viaja hasta los ribosomas donde la proteína será codificada.

ARNt El ARN de transferencia es una molécula de RNA que tiene asociado un aminoácido. En los ribosomas se asocian tres de las bases nitrogenadas del ARNm (denominado

[2] Para ser exactos habría que recordar que pueden existir secuencias de ADN denominadas plásmidos separadas del resto de ADN. Pero estos plásmidos no codifican proteínas a no ser que se incorporen al ADN de la célula.

[3] Excepto en algunos organismos como Cryptosporidium parvum que carecen de ADN mitocontrial http://www.cell.com/trends/parasitology/abstract/S1471-4922(04)00296-X

codón) con las del ARNt complementario portador del aminoácido asociado a esas tres bases nitrogenadas.

ARNr El ARN ribosómico es el que se encuentra dentro de los ribosomas. Su misión es la de proporcionar un mecanismo por el cual el ARNt se pueda hibridar con el ARNm.

A.4 Síntesis de proteínas

La principal función del *ADN* es la de codificar las proteínas que posteriormente serán sintetizadas en el interior de los ribosomas. El *dogma central de la biología molecular*[2] explica este proceso. Las proteínas son moléculas orgánicas formadas por la unión de varios aminoácidos. La secuencia de ADN que codifica una proteína la llamaremos *gen*, aunque existen varias definiciones sobre lo que es un gen nosotros nos limitaremos únicamente a esta[103].

El proceso de creación de una proteína comienza con la transcripción del ADN en ARNm. La información del ADN se encuentra codificada en grupos de tres nucleótidos llamados codones. Hay 4^3 posibles codones para codificar los 20 posibles aminoácidos y las señales que indican donde empieza y acaba la proteína.

Empezando a leer desde la terminación 5' a la 3' existen seis diferentes posiciones por donde empezar a leer una cadena (tres bases del codón junto con su correspondiente secuencia complementaria la otra cadena y en el otro sentido). El marco de lectura abierto [4] esta formado por los nucleótidos desde su comienzo hasta la señal de parada que codifican una proteína.

Una vez que el marco de lectura abierto es conocido la secuencia de ADN puede ser traducida en la secuencia de aminoácidos correspondiente. Esta secuencia siempre empieza por ATG en la mayoría de las especies y termina con TAA, TAG o TGA[5]

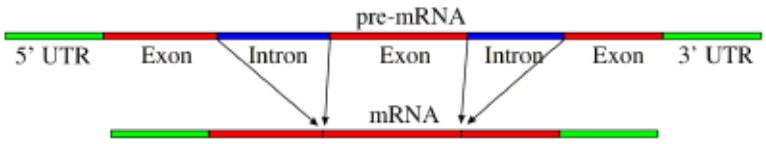


Figura A.2: Creación de ARNm en eucariotas. Fuente: Wikipedia.

La secuencia de ADN es transcrita a una cadena de ARNm siguiendo un proceso diferente en eucariotas y en procariotas. Dentro de un marco de lectura en las eucariotas el ADN contiene secuencias que no codifican ningún aminoácido. Esta región se denomina exón a diferencia del intrón que es la parte que codifica el aminoácido y es copiada al ARNm. Así que en las eucariotas el ARNm no es una copia directa del ADN, sino una copia del ADN quedándose sólo con los exones como se muestra en la figura A.2. La secuencia de DNA que codifica la proteína, sin intrones, es denominada **ADNc** (ADN complementario). Secuencias de ADNc son las que se utilizan en los microarrays para detectar el nivel de expresión de ciertos genes tal y como se explica en la sección A.5.1.

[4] También conocido como ORF (Open Reading Frame)

[5] http://bioweb.uwlax.edu/GenWeb/Molecular/Seq_Anal/Translation/translation.html

Una vez que la secuencia de ARNm es generada se desplaza hasta los ribosomas donde el RNAt se encarga de transportar los aminoácidos que le corresponden a cada codón. Los aminoácidos se van uniendo entre sí y van generando la proteína codificada en la secuencia de RNAm tal y como se muestra en la figura A.3.

Se dice que un gen está expresado si ese gen está sintetizando proteínas, lo que significa que hay moléculas de ARNm asociadas a ese gen en la célula. Es la concentración de esas moléculas de ARNm lo que permitirá saber cuanto de expresado está un gen.

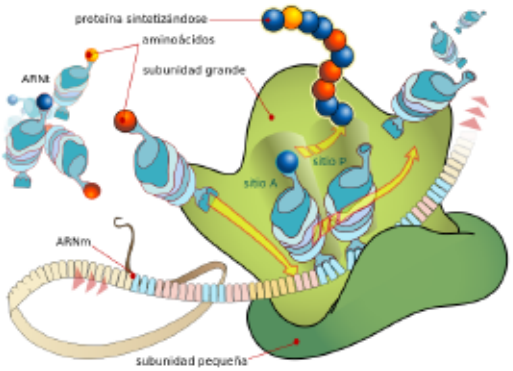


Figura A.3: Funcionamiento de un ribosoma. Fuente: Wikipedia.

A.5 Matriz de expresión genética

Como se dijo en la sección anterior cuando un gen se encuentra activo, está sintetizando la proteína que codifica, se pueden encontrar muestras de ARNm de ese gen en la célula. Si lo que se desea es saber como de expresado está un gen lo que hay que hacer es obtener la concentración de las cadenas de ARNm que se desean estudiar.

Existen principalmente dos técnicas para obtener la concentración de ARNm en las células, una basada en microarrays, donde se confía en que las cadenas de ARNm conocidas se hibridicen con unas sondas colocadas a tal propósito, y otra basada en técnicas RNA-Seq donde directamente se secuencian todas las cadenas de ARNm encontradas y se cuentan.

La ventaja del RNA-Seq frente a los microarrays es que el nivel de ruido es mucho más bajo, un problema muy común que tienen los microarrays es el ruido de fondo causado por la hibridación no deseada, además parece que para niveles de baja o muy alta expresión no ofrecen resultados muy fiables. También se reduce el número de análisis que hay que hacer sobre el mismo tejido para evitar posibles errores en el proceso[94]. Además los microarrays sólo permiten obtener el nivel de expresión de genes ya conocidos, mientras que con RNA-Seq no solo se pueden detectar niveles de expresión de genes desconocidos sino que además se pueden detectar pequeñas mutaciones en los mismos si las hubiera. Por contra hoy en día el análisis mediante microarrays es más barato y más sencillo, pero se espera que poco a poco esto deje de ser un problema y RNA-Seq se acabe imponiendo frente al uso de microarrays[104][4].

A.5.1 Microarray

Los microarrays (o chips de ADN) que permiten analizar miles de genes a la vez.

Cada microarray está formado por miles de sondas de hibridación. Cada *sonda*(*probeset*) de estas consiste en una secuencia de longitud variable de ADN o ARN que se asocia a una cadena complementaria (generalmente parte de una secuencia conocida de ARNm).

El proceso, muy resumido, que se sigue para obtener la expresión de un gen a partir de una muestra orgánica es [6] :

1. Obtención de una muestra del tejido que se desea analizar.
2. Purificación y aislamiento del ARN. Mediante un disolvente y una centrifugadora se consigue separar el ARN del resto de componentes de la muestra.
3. Se hace una copia del ARN mediante ADNc que tiene un tinte fluorescente.
4. Se deposita la muestra con ADNc fluorescente sobre el microarray. En este momento se producirá la hibridación entre las cadenas de ADNc y las sondas predefinidas del microarray. Una vez finalizada la hibridación se lava para eliminar las cadenas de ADNc que quedan libres.
5. Se escanea el microarray. Gracias al tinte fluorescente las sondas donde se ha producido la hibridación aparecen con mayor intensidad. En función de la intensidad con que emite cada sonda se calcula el nivel de expresión de cada gen.
6. Analizar los datos. En este punto se centra esta tesis.

Existen microarrays de uno o de dos canales. Los de dos canales utilizan dos tintes diferentes para analizar dos muestras independientes a la vez, el valor de interés es la relación que hay entre cada uno de los tintes que indican como de sobre-expresado o infra-expresado está el gen en cuestión en una muestra respecto a la otra. Los de un canal utilizan solo un tinte y en cada microarray se analiza una única muestra. En este caso hay que comparar varios microarrays para tener una idea de como de sobre o infra-expresado está un gen.

En los experimentos con microarrays siempre nos interesa en nivel relativo de una muestra respecto a otra. Los valores absolutos carecen de importancia porque un gen que en valor absoluto se encuentre más expresado que otro no significa que tenga una mayor importancia en el funcionamiento celular o que se vayan a sintentizar más proteínas. Por eso estos experimentos se centran en ver como cambia la expresión de ciertos genes de un experimento a otro.

A.5.1.1 Affymetrix

De los múltiples tipos de microarrays que existen, en esta tesis nos hemos centrado en los de Affymetrix[7] por ser los más utilizados y de los que más datos se encuentran publicados. Affymetrix es una empresa que se dedica a fabricar microarrays para diversos organismos. En esta tesis se han utilizado varios, ATH1 para *Arabidopsis Thaliana* y HG-U133Plus2, HG-U133A, Primeview para *Homo Sapiens*.

Después de varios pasos que incluyen purificación de las muestras biológicas, amplificación (por PCR), hibridación y escaneo del microarray, una imagen es obtenida donde el nivel de activación de cada sonda es revelado. De esta imagen se genera un fichero CEL [8], pero estos

[6] http://learn.genetics.utah.edu/content/labs/microarray/
[7] http://www.affymetrix.com/
[8] http://www.stat.lsa.umich.edu/ kshedden/Courses/Stat545/Notes/AffxFileFormats/cel.html

ficheros necesitan un procesado para poder mostrar el nivel de expresión de los genes. Este proceso es repetido varias veces y para cada ensayo se acaba con varios archivos CEL, cada uno se corresponde con una muestra diferente. Comparando el resultado entre diferentes archivos CEL es como se pueden sacar conclusiones acerca de la expresión de los genes.

Para obtener los valores de expresión de los genes en el array es necesario aplicar un algoritmo a los ficheros CEL con ayuda del Chip Description File (CDF) proporcionado por Affymetrix. Este fichero CDF define las reglas de cómo las sondas están relacionadas y a que gen corresponden. Existen varios algoritmos de preprocesado para hacer comparables como podrán ser el *Micro Array Suite 5.0* (MAS5.0) [105] o RMA[66].

A.5.1.2 Probesets y genes

Cada microarray recoge el nivel de concentración de diversas cadenas de ARNm. La combinación de varios microarrays, uno por cada experimento, genera la matriz de expresión genética que se desea analizar. Pero los microarrays no buscan directamente cadenas enteras de ARNm sino que tratan de hibridizar con subcadenas de decenas de pares de bases dentro de cada cadena de ARNm, varias de estas subcadenas se combinan y generan lo que se denomina un probeset. Una cadena de ARNm puede ser analizada por varias probesets y a su vez varias probesets se pueden hibridizar con una misma cadena de ARNm. Existen diferentes alternativas para, una vez obtenida la matriz de expresión de probesets, obtener el nivel de expresión de los genes, en [66][67] se ofrecen varias alternativas de cómo realizar este paso. En la sección 3.2.1.1 se explican estas técnicas.

A.5.2 RNA-Seq

Los métodos de secuenciación de ADN han existido desde hace décadas, quizá el más popular de todos ellos haya sido el descrito por Sanger[106] y en el cual se basan muchos métodos más modernos de secuenciación. El precio de secuenciar una cadena de ADN ha ido bajando con los años (figura A.4), al principio siguiendo la ley de Moore pero recientemente ha mostrado un abaratamiento aún más pronunciado gracias a nuevas técnicas de secuenciación que se han venido a denominar Next-Generation Sequencing (NGS)[107][108].

Estas nuevas técnicas permiten secuenciar genomas enteros, ver iteraciones entre ADN y las proteínas y el caso que nos interesa que es el estudio del transcriptoma que se denomina RNA-Seq[109].

El funcionamiento exacto de secuenciación del ARNm varía en función de la técnica utilizada, pero se podría decir que todas las técnicas RNA-Seq comparten ciertos puntos a la hora de analizar un tejido y se podrían resumir de la siguiente forma: Primero, para poder obtener el nivel de expresión de los distintos genes hay que convertir las cadenas de ARNm en ADN complementario (ADNc). Después se secuencian todos los fragmentos de las cadenas de ADNc, generalmente no se secuencia una cadena entera de ADNc correspondiente a un gen entero sino varios fragmentos del mismo. Esto genera una gran cantidad de fragmentos que tienen que ser identificados utilizando un *genoma de referencia* con los genes etiquetados. Por último con el *genoma de referencia* y los fragmentos de ADNc secuenciados se consigue una

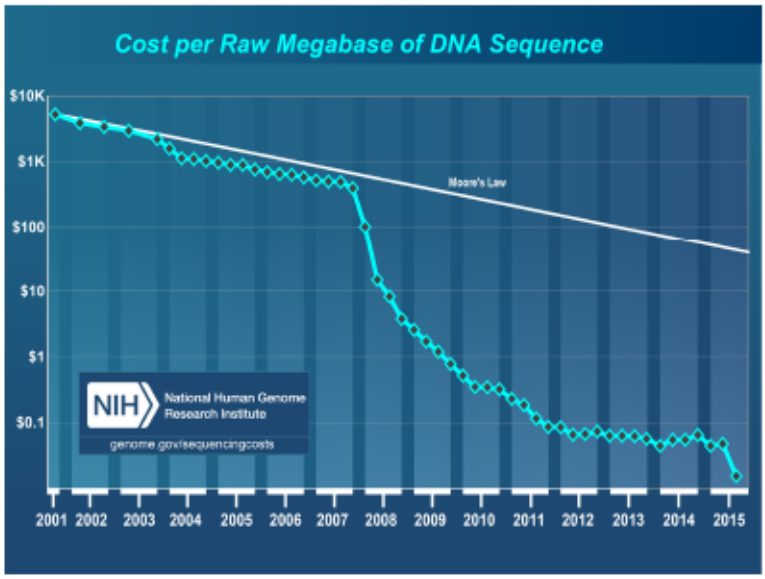


Figura A.4: Coste en dólares de secuenciación de una cadena de un millón de bases de ADN en relación al año. Fuente: http://www.genome.gov/sequencingcosts

tabla donde se muestra el número de veces que una secuencia de ARNm (o su complementario ADNc) aparece expresada.

A continuación se muestra con más detalle los pasos que hay que seguir para conseguir la tabla de conteos de ARNm partiendo de la lista de fragmentos de ADNc que se han podido secuenciar.

A.5.2.1 Análisis

La salida de un experimento de RNA-Seq son varios ficheros con todas las cadenas de ADNc secuenciadas. Cada muestra analizada produce un fichero en formato FASTQ[110] que es un fichero en formato texto dividido en grupos de 4 líneas, cada uno asociado con una cadena de ADN secuenciado. A continuación se muestra un fragmento de un fichero FASTQ real:

```
@SRR031712.1 HWI-EAS299\_10:4:1:1089:1668 length=45
AAATCGGGTAACTAAGCCAAAGAAAATATTCAAGGGAAGTTCAAA
+SRR031712.1 HWI-EAS299\_10:4:1:1089:1668 length=45
:::::::::::::::)\&7::7:,229277711133,.)(,\&\%-\&\%
```

La primera línea comienza con el carácter @ y contiene una descripción opcional de la secuencia. La segunda línea es la cadena secuenciada donde cada carácter corresponde a un nucleótido. La ternera línea comienza con + y contiene otra descripción opcional. La cuarta línea indica la calidad de la secuencia de la línea 2, aquí cada carácter tiene asociado un valor de calidad que indica cómo de probable es que la lectura del nucleótido sea errónea, se suelen utilizar los caracteres ASCII del 33 al 126.

Es muy probable que el los datos del experimento no se encuentren en formato FASTQ sino en Sequence Read Archive (SRA), estos dos tipos de ficheros son equivalentes, y se

puede convertir un fichero .sra en .fastq con ayuda del SRA Tookit[9]:

```
fastq-dump --split-3 file.sra
```

Estas secuencias de ADN deben ser alineadas y mapeadas a un *genoma de referencia*. De esta forma se puede saber a qué genes pertenecen los fragmentos de ADN secuenciados, además se puede contar el número de secuencias que aparecen para cada gen lo que permite obtener una estimación del nivel de expresión.

El *genoma de referencia* puede aparecer en dos ficheros separados. El primero aparece en formato FASTA y es el que contiene la secuencia completa del genoma en modo texto. Las líneas que comienzan por el carácter > muestran una descripción de la secuencia que viene a continuación distribuida en varias líneas. Este fichero se puede bajar de Ensembl http://www.ensembl.org/info/data/ftp/index.html. A continuación se muestran cuatro líneas de ejemplo de un fichero FASTA correspondiente al genoma de la *Drosophila Melanogaster* concretamente a la primera parte del cromosoma 2L:

```
>2L dna:chromosome chromosome:BDGP5:2L:1:23011544:1 REF
CGACAATGCACGACAGAGGAAGCAGAACAGATATTTAGATTGCCTCTCATTTTCTCTCCC
ATATTATAGGGAGAAATATGATCGCGTATGCGAGAGTAGTGCCAACATATTGTGCTCTTT
GATTTTTTGGCAACCCAAAATGGTGGCGGATGAACGAGATGATAATATATTCAAGTTGCC
```

Para un acceso más rápido a la secuencia genética se suele construir un índice de referencia a partir del fichero FASTA mediante la herramienta bowtie[10]:

```
bowtie2-build -f genome.fa genome_index
```

Además para el análisis de la secuencia hace falta un fichero GTF (Gene Transfer Format) el cual se utiliza para almacenar información de la estructura de genes. Es un fichero de texto donde cada campo dentro de una misma línea está separado por tabuladores y contiene información como identificador del gen y lugar donde se encuentra dentro de cada cromosoma. Más información sobre el formato de este fichero se puede consultar en http://mblab.wustl.edu/GTF22.html, también está disponible para descarga en Ensembl para diferentes organismos.

```
2L   protein_coding   exon   1954314   1954519   .   -   .
gene_id "FBgn0031375"; transcript_id "FBtr0330201";
exon_number "1"; gene_name "erm"; gene_biotype "protein_coding";
transcript_name "erm-RC"; exon_id "FBgn0031375:2";
```

A partir de los fichero FASTQ que contienen las secuencias detectadas y junto con los ficheros del genoma FASTA y GTF se genera un nuevo fichero en formato texto con extensión SAM[111] o su equivalente comprimido binario BAM donde aparecen todas las secuencias de ADNc alineadas al genoma de referencia. Esto se puede realizar con ayuda de la aplicación tophat2:

[9] http://www.ncbi.nlm.nih.gov/
[10] http://bowtie-bio.sourceforge.net

```
tophat2 -G genome.gtf -p 5 -o OutputFile genome_index  file.fastq
```

Una vez que se tienen los ficheros SAM se puede realizar los conteos, calcular cuantas veces aparece cada cadena de ARNm que pertenece a un gen. Esto se puede realizar con la función *htseq-count*:

```
htseq-count -s no -a 10 Output.sam genome.gtf > output.count
```

Este fichero de conteos será una columna de la matriz de expresión genética a partir de la cual se podrán realizar los estudios del trasncriptoma deseados. Una descripción más detallada de como obetner la matriz de conteos a partir de los ficheros FASTA y FASTQ de cada experimento se puede encontrar en [112].

B

Conceptos de Álgebra de Semianillos

B.1 Orden

Una *relación de orden (parcial)* en un conjunto P es una relación binaria $\leq$ que cumple las siguientes propiedades[1]:

(i) **reflexiva**: $x \leq x$

(ii) **antisimétrica**: $x \leq y$ y $y \leq x$ implica $x = y$

(iii) **transitiva**: $x \leq y$ y $y \leq z$ implica $x \leq z$

Un *conjunto parcialmente ordenado (conjunto parcialmente ordenado (cpo))* es el álgebra $\mathcal{P} = \langle P, \leq \rangle$ en donde P es el conjunto que porta una relación de orden $\leq$ [2]. El *dual de un cpo* $\mathcal{P}$ se representa con la notación $\mathcal{P}^\delta$ y es simplemente el conjunto original P con el orden invertido o dual $x \leq^\delta y \iff y \leq x$. La mayoría de las veces escribimos el orden invertido cambiando la dirección del símbolo $x \leq^\delta y \iff x \geq y$.

Si en un conjunto ordenado existe un elemento que es mayor que el resto de elementos diremos que es el *supremo (ing. "top")*, se representa con el símbolo $\top$, y cumple $x \leq \top \ \forall x \in P$. De la misma forma, podemos definir el *ínfimo del conjunto (ing. "bottom")* que se representa como $\bot$ y, de existir, cumple que $\bot \leq x \ \forall x \in P$.

Por ejemplo el conjunto $\wp(X)$, que es el conjunto formado por todos los subconjuntos de X, está ordenado por inclusión y por la inclusión inversa:

$$\forall A, B \in \wp(X) \quad A \leq B \iff A \subseteq B \quad B \leq^\delta A \iff B \supseteq A$$

Es fácil ver que, en el orden de inclusión, el supremo es el propio $\top = X$ mientras que el ínfimo es $\bot = \varnothing$. En el orden dual estos papeles se invierten.

Un conjunto ordenado donde todos sus elementos son comparables entre sí se denomina un *orden total, o cadena*. Por ejemplo, el conjunto ordenado de números reales $\mathbb{R}$ es una cadena sin supremo ni ínfimo. El intervalo $[2, 7]$ de los números naturales $\mathbb{N}$ es una cadena con $\bot = 2$ y $\top = 7$.

Una *anticadena* es un conjunto ordenado en el que se cumple $x \leq y$ solo si $x = y$.

[1] En esta sección seguimos a [113].

[2] En esta tesis usamos letras caligráficas para denotar álgebras y mayúsculas rectas para designar conjuntos.

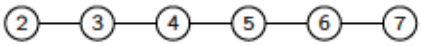


Figura B.1: Ejemplo de una cadena

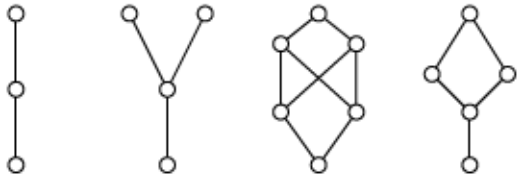


Figura B.2: Diversos diagramas de Hasse

B.1.1 Funciones sobre cpos

Una función entre dos conjuntos ordenados $\mathcal{P}$ y $\mathcal{Q}$ definida como $\varphi : P \to Q$ puede ser:

(i) **monótona**: si $x \leq_{\mathcal{P}} y$ en P entonces $\varphi(x) \leq_{\mathcal{Q}} \varphi(y)$ en Q.

(ii) **antítona**: si $x \leq_{\mathcal{P}} y$ en P entonces $\varphi(x) \geq_{\mathcal{Q}} \varphi(y)$ en Q; equivalentemente, si ϕ es una función monótona entre $\mathcal{P}$ y $\mathcal{Q}^{\delta}$.

(iii) **encaje (ing. "embedding")**: $\varphi : P \hookrightarrow Q$ si y sólo si $x \leq_{\mathcal{P}} y$ entonces $\varphi(x) \leq_{\mathcal{Q}} \varphi(y)$.

(iv) **isomórfica** : si además de cumplir las premisas anteriores es una función biyectiva. Esto implica que tiene inversa $\varphi^{-1} : Q \to P$ la cual es a su vez isomórfica. Dos conjuntos isomórficos son esencialmente indistinguibles por la relación de orden.

(v) **anti-isomórfica** : Un anti-isomorfismo entre $\mathcal{P}$ y $\mathcal{Q}$ es un isomorfismo entre $\mathcal{P}$ y $\mathcal{Q}^{\delta}$.

B.1.2 Diagrama de Hasse

La forma gráfica de representar un conjunto ordenado es mediante un *diagrama de Hasse o diagrama de orden*. En estos diagramas bidimensionales cada elemento se muestra en un vértice mientras que los arcos que unen los vértices indican la relación de orden que hay entre los elementos. Si $x < y$ el vértice de x debe situarse por debajo de y. Los arcos pueden cruzarse entre sí pero no pueden tocar ningún vertice diferente de su origen y final. En la Figura B.2 se muestran algunos diagramas de Hasse:

B.1.3 Filtro e ideal

Un subconjunto Q dentro de un conjunto ordenado $(P, \leq)$ es un *ideal (de orden)* si se cumple:

$$\forall x \in Q \;\; y \leq x \Rightarrow y \in Q \tag{B.1.1}$$

Definimos $\downarrow Q$ como el *ideal* más pequeño que contiene Q, donde:

$$\downarrow Q := \{y \in P \mid (\exists x \in Q)\, y \leq x\}\,. \tag{B.1.2}$$

Entonces Q es un subconjunto ideal de P si y sólo si $Q = {\downarrow} Q$. El subconjunto ideal más pequeño que contiene el elemento x se llama el *ideal principal (de x)* y se define como ${\downarrow} x := \{y \in P \mid y \leq x\}$.

El concepto dual al de ideal es el de *filtro (de orden)*, que es un subconjunto $Q \subseteq P$ tal que

$$\forall x \in Q \;\; x \leq y \Rightarrow y \in Q\,. \tag{B.1.3}$$

Al igual que en el caso anterior podemos decir que el subconjunto Q es un *filtro* de P si y sólo si $Q = {\uparrow} Q$, donde:

$$\uparrow Q := \{y \in P \mid (\exists x \in Q)\, y \geq x\}\,. \tag{B.1.4}$$

El *filtro principal (de p)* se define igualmente como $\uparrow p := \{y \in P \mid y \geq p\}$.

Sea una aplicación $f : P \to Q$, se dice que es *residuada (ing. "residuated")* si existe una función (que es un *residuo*, ver más abajo) $f^{\#} : Q \to P$ definida por[114, p.172]:

$$f^{\#}(q) = \bigvee \{p \in P | f(p) \leq q\} \tag{B.1.5}$$

Esta definición implica que f es residuada si la pre-imagen de cada ideal principal de Q es a su vez un ideal principal de P.

Sea una función $g : Q \to P$, se dice que es un *residuo (ing. "residual")*, si existe una función (que es residuada, ver más arriba), $g^{\flat} : P \to Q$, definida por,

$$g^{\flat}(p) = \bigwedge \{q \in Q | p \leq g(q)\} \tag{B.1.6}$$

Esto implica que g sea un residuo si la pre-imagen de cada filtro principal de Q es a su vez un filtro principal de P. Nótese que los conceptos de aplicación residuada y residual refinan el de función monótona sobre un orden (ver. B.1.1).

Las composiciones de aplicaciones residuales y residuadas generan operadores expansivos y contractivos. Por ejemplo, con las definiciones de arriba, se tiene que:

$$f\left(f^{\#}(q)\right) \leq q \qquad\qquad g\left(g^{\flat}(p)\right) \geq p \tag{B.1.7}$$

Además, f y $f^{\#}$, resp. g y $g^{\flat}$ son pseudoinversas:

$$f^{\#}\left(f\left(f^{\#}(q)\right)\right) = f^{\#}(q) \qquad\qquad g^{\flat}\left(g\left(g^{\flat}(q)\right)\right) \geq g^{\flat}(q) \tag{B.1.8}$$

B.2 Retículos

En esta sección profundizamos en la estructura de los cpos de la Sección B.1 y de las operaciones factibles sobre sus elementos, que nos llevarán al concepto de retículo.

B.2.1 Cotas superiores e inferiores, supremos e ínfimos

Sea un cpo $\mathcal{P}$. Consideremos $Q \subseteq P$ y llamemos:

- *cota o límite superior de* Q a $x \in P$ tal que $\forall q \in Q, q \leq x$.
- Dualmente, *cota o límite inferior de* Q a $y \in P$ tal que $\forall q \in Q, q \geq y$.

El conjunto de todos los límites superiores de Q es un *filtro* de P y se define como:

$$Q^u := \{x \in P \mid (\forall q \in Q)\, q \leq x\}$$

Si Q^u tiene un elemento inferior al resto, entonces se denomina *supremo o mínima cota superior* de Q,

$$\sup Q = \vee Q = !x \in P \quad \text{tal que} \quad \forall p \in P, [((\forall q \in Q)\, q \leq p) \Rightarrow x \leq p]$$

A su vez, el conjunto de todos los límites inferiores de Q es un *ideal* de P y se define como:

$$Q^l := \{y \in P \mid (\forall q \in Q)\, q \geq y\}$$

De forma dual al caso anterior si Q^l tiene un elemento superior al resto se define como *ínfimo*,

$$\text{ínf } Q = \wedge Q = !x \in P \quad \text{tal que} \quad \forall p \in P\, [((\forall q \in Q)\, q \leq p) \Rightarrow y \geq p]$$

En la Figura B.3 se muestra el conjunto ordenado P y el subconjunto Q junto con Q^l y Q^u.

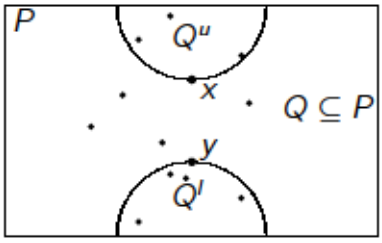


Figura B.3: Subconjunto ordenado Q junto con Q^u y Q^l

Si tenemos un conjunto ordenado P donde existen los elementos $\top$ y $\bot$ como se definen en la sección B.1, es fácil ver que $P^u = \{\top\}$ y que por lo tanto el *supremo* de P es $\vee P = \top$. Por dualidad se puede demostrar que el *ínfimo* de P es $\wedge P = \bot$.

B.2.2 Operador $\vee$ y $\wedge$

Un cpo $\mathcal{L}$ es un *retículo (ing. "lattice")*, si cada par de elementos $x, y \in L$ tiene supremo $x \vee y \in L$ e ínfimo $x \wedge y \in L$. Claramente, los operadores $\vee$ y $\wedge$ son funciones en $L^2 \to L$ que preservan el orden. Un cpo P es *completo (como orden)* [115, def.4.1] si para cada subconjunto $A \subseteq P$, existe tanto el supremo como el ínfimo de A. Nótese que todos los

conjuntos parcialmente ordenados completos son retículos, y un retículo L que es completo como un conjunto parcialmente ordenado es un *retículo completo*.

Un *semi-retículo superior (ing. "join-semilattice, upper-semilattice")* es un cpo con un supremo para cualquier subconjunto finito no vacío. A su vez un *semirretículo inferior (ing., "meet-semilattice, lower-semilattice)* se define de forma dual como un cpo con un ínfimo para cualquier subconjunto finito no vacío. Entonces, un retículo es un cpo que es a la vez un semirretículo superior e inferior.

Por ejemplo en Figura B.4 se puede ver que $a \vee c = a$ y que $a \vee b = \top$. En cambio $c \vee d$ no existe ya que $\{c, d\}^u = \{\top, a, b\}$. De una forma similar $a \wedge c = a$ y $c \wedge d = \bot$, en este caso $a \wedge b$ no existe ya que $\{a, b\}^l = \{\bot, c, d\}$.

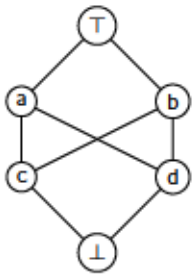


Figura B.4: Diagrama de Hasse de un conjunto parcialmente ordenado.No se trata de un retículo ya que no todos los pares de elementos tienen un supremo e ínfimo definidos.

Supremo e ínfimo (binarios) son operadores internos, $\forall a, b \in M \subseteq L$, $a \vee b \in M$ y $a \wedge b \in M$ que además cumplen las leyes:

- *asociativa*: $(a \vee b) \vee c = a \vee (b \vee c) \quad (a \wedge b) \wedge c = a \wedge (b \wedge c)$
- *idempotente*: $a \vee a = a \vee a \quad a \wedge a = a \wedge a$
- *conmutativa*: $a \vee b = b \vee a \quad a \wedge b = b \wedge a$
- *absorción*: $a \wedge (a \vee b) = a \quad a \vee (a \wedge b) = a$

Entre grupos ordenados, y por extensión en retículos, se puede definir un producto cartesiano imponiendo un orden para cada una de las coordenadas. Así pues si tenemos 2 retículos $L_1 \times L_2$ su producto cartesiano será:

$$(a_{l1}, a_{l2}) \vee (b_{l1}, b_{l2}) = (a_{l1} \vee b_{l1}, a_{l2} \vee b_{l2})$$
$$(a_{l1}, a_{l2}) \wedge (b_{l1}, b_{l2}) = (a_{l1} \wedge b_{l1}, a_{l2} \wedge b_{l2})$$

La forma de dibujar este producto consiste en reemplazar cada punto del diagrama de L_1 por L_2 y conectar los puntos tal y como se muestra en la figura B.5. Es fácil ver que el producto de dos retículos siempre contiene subretículos isomórficos a cada uno de los retículos originales.

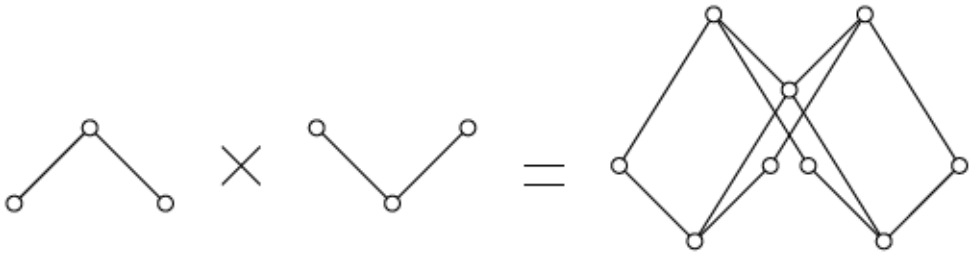


Figura B.5: Producto cartesiano de dos órdenes.

B.2.3 Funciones

Una función entre dos retículos $L \to K$ se dice que es *homomórfica o un homomorfismo de retículos* si preserva los supremos e ínfimos $a, b \in L$:

$$f(a \vee b) = f(a) \vee f(b)$$

$$f(a \wedge b) = f(a) \wedge f(b)$$

Un homomorfismo biyectivo de retículos es un *isomorfismo (de retículos)*.

En el apartado B.1.3 se explicó la definición *ideal* y *filtro* para cpo. Un subconjunto Q de un retículo L es un *ideal (de retículo)* si además es un un conjunto cerrado bajo el operador de supremos:

$$a, b \in Q \to a \vee b \in Q$$

$$a \in L, b \in L \quad a \leq b \to a \in L$$

De forma dual un subconjunto Q dentro de un retículo L es un *filtro (de un retículo)* si además es un un conjunto cerrado bajo el operador de ínfimos:

$$a, b \in Q \to a \wedge b \in Q$$

$$a \in L, b \in L \quad a \geq b \to a \in L$$

El conjunto de todos los posibles ideales y filtros de un retículo L se representan por $\mathcal{I}(L)$ y $\mathcal{F}(L)$ y tienen el orden definido por la inclusión de subconjuntos.

B.2.4 Elementos irreducibles

En un espacio vectorial podemos tener un conjunto de vectores linealmente independientes cuya combinación lineal genera cualquier elemento dentro de dicho espacio. Este concepto se puede extender a los retículos definiendo elementos *irreducibles por la operación de tomar supremos o supremo- o $\vee$-irreducibles* e *irreducibles por la operación de tomar ínfimos o ínfimo- o $\wedge$-irreducibles*.

Un elemento x de un retículo L será supremo-irreducible si cumple:

$$x \neq 0 \quad \text{(En caso que L tenga un cero o "bottom")}$$
$$x = a \vee b \to x = a \quad \text{o} \quad x = b \quad \forall a, b \in L$$

Esto implica que en un retículo finito un elemento es supremo-irreducible si y sólo si tiene únicamente un vecino inferior inmediato. El conjunto formado por todos los elementos supremo-irreducibles de un retículo L se representa como $\mathcal{J}(x)$.

De forma dual, se definen los ínfimo-irreducibles:

$$x \neq 1 \quad \text{(En caso que L tenga un supremo o "top")}$$
$$x = a \wedge b \rightarrow x = a \quad \text{o} \quad x = b \quad \forall a, b \in L$$

en un retículo finito un elemento será ínfimo-irreducible si y solo si tiene un único elemento superior. El conjunto formado por todos los elementos ínfimo-irreducibles de un retículo L se representa como $\mathcal{M}(x)$.

Un conjunto Q es *supremo-denso* (ing. "join-dense") en L si cada elemento de L se puede obtener como el supremo de un subconjunto de elementos de Q [116] :

$$Q \subseteq L, a \in L \rightarrow a = \bigvee_{p \in P} A, A \subseteq Q$$

Dualmente se define un subconjunto como *ínfimo-denso* (ing. *meet-dense*) si cada elemento del conjunto del que procede se puede obtener mediante la intersección de elementos de dicho subconjunto.

Un resultado fundamental es que el conjunto de supremo-irreducibles de un retículo finito es supremo-denso en ese retículo y dualmente para los ínfimo irreducibles.

B.3 Conexión de Galois

Bajo el nombre de "conexión de Galois" se agrupan diversos tipos de pares de funciones entre dos cpos [113, 117]. A continuación vamos a estudiar dos casos particulares por su relevancia para el FCA.

B.3.1 Conexión covariante o adjunción

Una conexión de Galois covariante o adjunción entre dos cpos consiste en un par de funciones monótonas entre los conjuntos $\langle \pi_*, \pi^* \rangle$, con $\pi_* : P \rightarrow Q$ $\pi^* : Q \rightarrow P$, que cumplen que $\forall x \in P, y \in Q$, $\pi_*(x) \leq y \Leftrightarrow x \leq \pi^*(y)$. Nombraremos a π_* como el *adjunto inferior o coadjunto (de la conexión)* y π^* como *adjunto superior o, simplemente, adjunto*[3]. Un ejemplo de una conexión de Galois entre dos conjuntos parcialmente ordenados se puede ver en la Figura B.6.

Se puede demostrar que la función π^* es un residuo (ver B.1.3), lo que significa que su inversa preserva los filtros principales, lo que a su vez implica que preserva ínfimos (ing. "upper-semicontinuous"; ver B.4.3)

$$\pi^*(q_1) \wedge \pi^*(q_2) = \pi^*(q_1 \wedge q_2) \tag{B.3.1}$$

[3]Esta notación no es estandar, algunos autores intercambian π^* y π_*

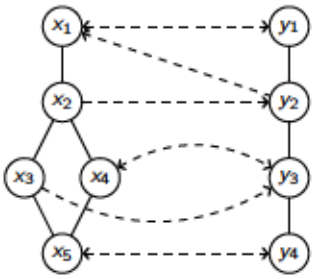


Figura B.6: Ejemplo de una conexión de Galois

A su vez la función π_* es una función residuada(ver B.1.3), lo que implica que su inversa preserva los ideales principales, es decir, que la función π_* preserva los máximos (lower-semicontinuous B.4.2):

$$\pi_*(p_1) \vee \pi_*(p_2) = \pi_*(p_1 \vee p_2) \tag{B.3.2}$$

De esta forma una función de una conexión de Galois determina a la otra. De la ecuación B.1.6 podemos sacar la función π_* y de (B.1.5) podemos obtener π^*:

$$\pi_*(p) = \bigwedge\{q \in Q | p \leq \pi^*(q)\} \qquad \pi^*(q)s = \bigvee\{p \in P | \pi_*(p) \leq q\} \tag{B.3.3a}$$

La función compuesta $\gamma = \pi^* \circ \pi_*$ es de hecho un *cierre u operador de cerrados (ing., "closure operator")*, es decir expansivo, idempotente e isótono [118]:

- expansivo: $x \leq \gamma(x)$
- idempotente: $\gamma(x) = \gamma(\gamma(x))$
- isótono: $x_1 \leq x_2 \rightarrow \gamma(x_1) \leq \gamma(x_2)$

De forma dual la función compuesta $\kappa = \pi_* \circ \pi^*$ es un *kernel u operador de abiertos* en P, es decir, contractivo, idempotente e isótono [119, cap.2]:

- contractivo: $\kappa(y) \leq y$
- idempotente: $\kappa(y) = \kappa(\kappa(y))$
- isótono: $y_1 \leq y_2 \rightarrow \kappa(y_1) \leq \kappa(y_2)$

B.3.2 Conexión contravariante

Como se dijo antes existe otra definición complementaria de una *conexión (contravariante) de Galois* que es igual a la anterior pero invirtiendo el orden del segundo conjunto [120], es decir, un par de funciones, $\lambda : P \rightarrow Q, \varrho : Q \rightarrow P$

$$p \leq \varrho(q) \qquad q \leq \lambda(p) \tag{B.3.4}$$

En este caso las dos funciones se llaman *polares* en lugar de adjuntos [117]. La relación entre los polares será:

$$\varrho(q) = \bigvee\{p \in P | \lambda(p) \leq q\} = \lambda^{\#}(q) \quad \lambda(p) = \bigvee\{q \in Q | \varrho(q) \leq p\} = \varrho^{\#}(p) \quad \text{(B.3.5)}$$

Como ambos operadores son mutuamente residuados, sus composiciones $\lambda \circ \varrho$ y $\varrho \circ \lambda$ son operaciones de cierre.

Cualquier relación $I \subseteq G \times M$ induce una conexión de Galois $P(G) \overset{R_+{}^+}{\to} P(M)^\delta$. Donde los componentes de $R_+{}^+ = \langle R_+, R^+ \rangle$ se definen como [117, prop.7]:

$$R_+(A) := \{m \in M | \forall g \in G \langle g, m \rangle \in R\} \text{ para } A \subseteq G \quad \text{(B.3.6)}$$
$$R^+(B) := \{g \in G | \forall m \in G \langle g, m \rangle \in R\} \text{ para } B \subseteq M \quad \text{(B.3.7)}$$

Esta conexión de Galois, descubierta por Ore [121] y Birkhoff [122] concurrente e independientemente [117], fue hecha operativa de nuevo por diversos autores [7, 123, 124] para la representación de datos que pertenecen a dos grupos diferentes y relacionados entre sí por medio de una matriz de incidencia. Por ejemplo, en [125] se estudia el caso de una serie de actores que coinciden en diferentes eventos, mediante una matriz booleana se define qué persona fue a qué evento. Esta matriz establece conexiones de Galois que permiten conocer con más de detalle el tipo de relación que puede existir entre ciertos actores. Como se explicará en detalle en la Sección 1.3 este tipo de estudio se conoce como Análisis en Conceptos Formales.

B.4 Teoría de Semianillos

Se puede probar que es posible generalizar la construcción de la conexión de Galois dada por una matriz booleana en la Sección B.3.2. Pero las estructuras algebraicas más abstractas que se conozca que permitan definir la conexión son los semianillos naturalmente ordenados o dioides. En esta sección, pues, exponemos brevemente la teoría de semianillos siguiendo fundamentalmente a [126, 127, 128].

B.4.1 Anillos y semianillos

El concepto clásico más relacionado con los semianillos es el de anillo. Un *anillo* es un conjunto equipado con dos operaciones binarias llamadas suma y multiplicación y se representaría como $\langle R, \oplus, \otimes, \epsilon, e \rangle$. Para ser considerado un anillo el conjunto R y las dos operaciones $\oplus$ y $\otimes$ tienen que satisfacer los siguientes axiomas [129, p.120] $\forall a, b, c \in R$:

(i) $a \oplus b \in R$

(ii) $a \oplus b = b \oplus a$

(iii) $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

(iv) $\exists \epsilon \in R \mid a \oplus \epsilon = a \quad \forall a \in R$

(v) $\exists - a \in R \mid a \oplus (-a) = 0$

(vi) $a \otimes b \in R$

(vii) $a \otimes (b \otimes c) = (a \otimes b) \otimes c$

(viii) $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ y $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$

(ix) $a \otimes e = e \otimes a = a$

Axiomas de *i* a *v* implican que $(R, \oplus)$ tiene que ser un grupo abeliano bajo la operación de suma. Los axiomas *vi* y *vii* dicen que R tiene que ser un semigrupo bajo el operador $\otimes$. El axioma *viii* muestra la ley distributiva. El axioma *ix* no es requerido por muchos autores en la definición de anillo. Aquí se supondrá que un anillo siempre tiene elemento unidad, es decir la operación de multiplicación define un monoide en el conjunto R.

Un anillo famoso es el conjunto de todos los enteros Z junto con la operación tradicional de suma y multiplicación y sus elementos neutros para cada operación sería 0 y 1 respectivamente, quedaría como $\mathbb{Z} = \langle Z, +, \cdot, 0, 1 \rangle$.

Un **semianillo** $\mathcal{S} = (S, \oplus, \otimes, \epsilon, e)$ es una generalización de un *anillo* que no tiene porqué cumplir el axioma *v*, es decir, que no exigen la existencia de inversos aditivos. Algunos autores [130, p.26] añaden un axioma más a los ya visto en la sección anterior obligando a que el elemento neutro de la suma ϵ sea absorbente bajo el operador de multiplicación $(\forall k \in K, \epsilon \otimes k = k \otimes \epsilon = \epsilon)$, y esta es la convención que seguiremos.

Ejemplos de semianillos son:

- el semianillo de los números enteros: $\mathbb{Z} = \langle Z, +, \cdot, 0, 1 \rangle$.
- el semianillo maxplus: $\mathbb{R}_{max,+} = \langle R \cup \{-\infty\}, \text{máx}, +, -\infty, 0 \rangle$.
- el semianillo minplus: $\mathbb{R}_{min,+} = \langle R \cup \{\infty\}, \text{mín}, +, \infty, 0 \rangle$.

En un semianillo $\mathcal{K}$ se pueden definir la multiplicación por la izquierda y por la derecha:

$$\begin{aligned} L_a : K &\to K & R_a : K &\to K \\ b &\mapsto L_a(b) = a \otimes bb & &\mapsto R_a(b) = b \otimes a \end{aligned}$$

Si la operación de multiplicación es conmutativa, entonces se dice que se trata de un *semianillo conmutativo* y se dice que es un *semicuerpo (ing. "semifield")* si su estructura multiplicativa es un grupo, es decir existe la operación inversa $\cdot^{-1} : K \to K$, tal que $\forall k \in K, k \otimes k^{-1} = k^{-1} \otimes k = e$.

B.4.2 Semianillos idempotentes

Un **semianillo idempotente** es un semianillo cuya operación de suma es idempotente $(\forall k \in K, k \oplus k = k)$. Todo lo monoide conmutativo e idempotente $\langle K, \oplus, \epsilon \rangle$ tiene implícita una relación de orden natural

$$\forall a, b \in K, a \leq b \iff a \oplus b = b \tag{B.4.1}$$

que lo convierte en un *semi-retículo superior* (*join-semilattice*) $\langle K, \vee, \bot \rangle$ tal y como se definió en B.2.1, donde el supremo se define como: $a \vee b = a \oplus b$. En este caso el elemento neutro para la suma será también el ínfimo del conjunto $\epsilon = \bot$.

En un conjunto ordenado $\langle R, \leq \rangle$, una función f es semicontinua inferior (*lower-semicontinuous*) si para cada subconjunto S de R se tiene que [131]:

$$f\left(\bigvee_{s \in S} s\right) = \bigvee_{s \in S} f(s) \tag{B.4.2}$$

De forma dual se define una función f semicontinua superior (*upper-semicontinuous*) si para cada subconjunto S de R:

$$f\left(\bigwedge_{s \in S} s\right) = \bigwedge_{s \in S} f(s) \tag{B.4.3}$$

Un semianillo idempotente $\mathcal{K}$ es **completo**, si es un conjunto parcialmente ordenado completo y la multiplicación por la izquierda (L_a) y por la derecha (R_a) son semicontinua inferior (*lower-semicontinuous*), es decir, son funciones residuadas (B.1.6). Es decir, un semianillo es completo si está definido el operador $\oplus$ para todos sus elementos y la multiplicación es distributiva sobre sumas finitas. La suma de todos los elementos de un semianillo completa será el elemento $\top$.

Tal y como demuestra en [114, Teorema 4.27] un *semi-retículo superior completo* es un retículo al poseer un elemento neutral para la suma, el elemento $\bot$. Por lo tanto un semianillo idempotente completo por definición es un semi-retículo completo tal y como se definió en B.2.1 con las operaciones de supremo e ínfimo definidas por las equivalencias:

$$\forall a, b \in K, a \leq b \iff a \vee b = b \iff a \wedge b = a \tag{B.4.4}$$

Un semianillo idempotente cuya estructura multiplicativa es un grupo es un *semicuerpo idempotente*. La fórmula del ínfimo en este caso, desarrollada por Dedekind, es: $a \wedge b = a^{-1} \otimes (a \oplus b) \otimes b^{-1}$ [77]. De esta forma al tener definidas las funciones de supremo e ínfimo los *semicuerpos idempotentes* son también retículos.

Por ejemplo el semianillo booleano $\mathcal{B} = \langle \mathbb{B}, \vee, \wedge, 0, 1 \rangle$ donde las operaciones están definidas como:

$$0 \vee 0 = 0 \qquad 0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$$

$$0 \wedge 1 = 1 \wedge 0 = 0 \wedge 0 = 0 \qquad 1 \wedge 1 = 1$$

es un semianillo idempotente, completo y conmutativo.

B.4.3 Semimódulos sobre semianillos idempotentes

El concepto de un módulo sobre un anillo es una generalización del concepto de espacio vectorial donde los escalares correspondientes residen en un anillo. Para un anillo R, M es un **módulo** sobre R (*ing. R-module*) si M es un grupo abeliano $\langle M, \oplus, \epsilon \rangle$ y tiene una operación $R \times M \to M$ de forma que $\forall x, x_1, x_2 \in M, \forall r, r_1, r_2 \in R$ [132, def.3.1]:

(i) $(r_1 \oplus r_2) \otimes x = r_1 \oplus x \oplus r_2 \oplus x$

(ii) $x \oplus (x_1 \oplus x_2) = (x \oplus x_1) \oplus x_2$

(iii) $e \otimes x = x$

Concretamente este *módulo* $\mathcal{M}$ es un módulo *por la izquierda* porque la operación de multiplicación está definida por la izquierda ($R \times M \to M$). De forma simétrica se puede definir un *módulo por la derecha* para el anillo $\mathcal{R}$ de forma que la multiplicación será una función del tipo $M \times R \to M$.

Se puede definir un **semimódulo** sobre un semianillo de la misma forma que se define un módulo sobre un anillo. Un $\mathcal{K}$-*semimódulo* por la izquierda, $\mathcal{M}$, de un *semianillo* $\mathcal{K}$ sigue las mismas reglas que las definidas para un módulo. Para este $\mathcal{K}$-*semimódulo* por la izquierda se definen las operaciones de multiplicación por la izquierda y por la derecha como:

$$\begin{aligned} L_k : M &\to M & R_m : K &\to M \\ m &\mapsto L_k(m) = k \otimes m & k &\mapsto R_m(k) = k \otimes m \end{aligned} \tag{B.4.5}$$

De forma dual se puede definir un $\mathcal{K}$-*semimódulo* $\mathcal{M}$ por la derecha con las operaciones de multiplicación por la izquierda y por la derecha como:

$$\begin{aligned} L_m : K &\to M & R_k : M &\to M \\ k &\mapsto L_m(k) = m \otimes k & m &\mapsto R_k(m) = m \otimes k \end{aligned} \tag{B.4.6}$$

Un ($\mathcal{K}$,$\mathcal{S}$)-*semimódulo* es un conjunto $\mathcal{M}$ con la estructura del $\mathcal{K}$-*semimódulo* por la izquierda y del $\mathcal{S}$-*semimódulo* por la derecha. Un $\mathcal{K}$-*bisemimódulo* tiene dos estructuras $\mathcal{K}$-*semimódulo*, por la izquierda y por la derecha, que conmutan.

Un $\mathcal{K}$-*semimódulo* por la izquierda $\mathcal{M}$ es completo si el semianillo $\mathcal{K}$ es **completo**, $\mathcal{M}$ es completo en el sentido de un conjunto parcialmente ordenado (B.2.1) y $\forall m \in \mathcal{M}, k \in \mathcal{K}$ las multiplicaciones por la izquierda L_k y R_m son ambas continuas. De forma simétrica se define un semimódulo por la derecha.

Un $\mathcal{K}$-*semimódulo* (de igual forma un $(\mathcal{K},\mathcal{K})$-*bisemimódulo*) $\mathcal{M}$ sobre un semianillo idempotente $\mathcal{K}$ hereda la ley idempotente la cual induce un orden natural como se muestra en B.4.4. Si además $\mathcal{M}$ es *completo* también es un retículo completo con las operaciones de supremo e ínfimo definidas como se explicó en B.2.2:

$$\forall a, b \in M, a \leq b \iff a \vee b = b \iff a \wedge b = a \tag{B.4.7}$$

Como ejemplo de $\mathcal{K}$-*semimódulos* obtenidos de [12]. Cada semianillo $\mathcal{K}$ es un semimódulo sobre el mismo. Así es un $(\mathcal{K},\mathcal{K})$-*bisemimódulo* sobre si mismo porque la acción de multiplicación conmuta sobre si misma. Tal es el caso de el $(\mathcal{B},\mathcal{B})$-*bisemimódulo* booleano y los bisemimódulos maxplus y minplus. Todos estos son completos e idempotentes.

El conjunto de matrices $K^{n\times p}$ para n y p finitos es un $(\mathcal{K}^{n\times n}, \mathcal{K}^{p\times p})$-*bisemimódulo*, así las operaciones por la izquierda y por la derecha consisten en la multiplicación de matrices:

$$K^{n\times n} \cdot K^{n\times p} \to K^{n\times p} \qquad K^{n\times p} \cdot K^{p\times p} \to K^{n\times p}$$

De igual forma el conjunto de vectores $K^{n\times 1}$ para n finito es un $(\mathcal{K}^{n\times n}, \mathcal{K})$-*bisemimódulo* con operaciones similares a la anterior:

$$K^{n\times n} \cdot K^{n\times 1} \to K^{n\times 1} \qquad K^{n\times 1} \cdot K \to K^{n\times 1}$$

Y por último, para acabar con este ejemplo, el conjunto de vectores $K^{1\times p}$ para p finito es un $(\mathcal{K}, \mathcal{K}^{p\times p})$-*bisemimódulo* con operaciones por la izquierda y derecha:

$$K \cdot K^{1\times p} \to K^{1\times p} \qquad K^{1\times p} \cdot K^{p\times p} \to K^{1\times p}$$

B.4.4 Residuación

En la sección B.3 se dio una definición de una función residuada y su papel en las conexiones de Galois. Ahora se ampliará realizando una definición más formal utilizando la teoría de semiódulos vistos en la sección anterior.

Si $\langle S, \leq\rangle$ y $\langle K, \leq\rangle$ son dos conjuntos ordenados completos entonces una función $f : S \to T$ es residuada si y sólo si es continua. En este caso existirá una función $f^{\#} : T \to S$ tal que:

$$f(s) \leq k \iff s \leq f^{\#}(k)$$

Donde $f^{\#}(k)$ será su *residuo* y al igual que en B.1.5, se define como:

$$f^{\#}(k) = \bigvee\{s \in S | f(s) \leq k\} \tag{B.4.8}$$

De forma simétrica podemos definir f como:

$$f(s) = \bigwedge\{k \in K | f^{\#}(k) \leq s\} \tag{B.4.9}$$

La función $f^{\#}$ es semicontinua superior (*ing. üpper-semicontinuous"*) ya que cumple la ecuación B.4.3, a su vez f será semicontinua inferior (*ing. "lower-semicontinuous"*) al cumplir B.4.2. Entre estas dos funciones se establece una relación de Galois como se vio en la sección B.3.1.

En un $\mathcal{K}$-semimódulo complete por la izquierda, $\mathcal{M}$, se puede definir el residuo de L_k y R_m aplicando B.4.8 en B.4.5 de forma que $\forall m_1, m_2 \in \mathcal{M}$ y $k \in \mathcal{K}$:

$$\begin{aligned} &L_k^{\#} : M \to M \\ &L_k^{\#}(m_1) = k \setminus m_1 = \bigvee\{m_2 \in M | k \otimes m_2 \leq m_1\} \end{aligned} \tag{B.4.10}$$

$$\begin{aligned} &R_m^{\#} : M \to K \\ &R_{m_2}^{\#}(m_1) = m_1 / m_2 = \bigvee\{k \in K | k \otimes m_2 \leq m_1\} \end{aligned} \tag{B.4.11}$$

Se puede establecer la siguiente relación entre B.4.10 y B.4.11:

$$k \otimes m_2 \leq m_1 \Leftrightarrow k \leq m_1 / m_2 \Leftrightarrow m_2 \leq k \setminus m_1 \tag{B.4.12}$$

Para un $\mathcal{K}$-*semimódulo* por la derecha completo, $\mathcal{M}$, bastaría con aplicar B.4.8 en B.4.6:

$$\begin{aligned} &L_m^{\#} : M \to K \\ &L_{m_2}^{\#}(m_1) = m_2 \setminus m_1 = \bigvee\{k \in K | m_2 \otimes k \leq m_1\} \end{aligned} \tag{B.4.13}$$

$$\begin{aligned} &R_k^{\#} : M \to M \\ &R_k^{\#}(m_1) = m_1/k = \bigvee\{m_2 \in M | m_2 \otimes k \leq m_1\} \end{aligned} \tag{B.4.14}$$

Al igual que en el caso del semimódulo por la izquierda se puede establecer la siguiente relación entre B.4.13 y B.4.14:

$$m_2 \otimes k \leq m_1 \Leftrightarrow k \leq m_2 \setminus m_1 \Leftrightarrow m_2 \leq m_1/k \tag{B.4.15}$$

Estas operaciones se pueden visualizar fácilmente con la ayuda del ejemplo 4.66 extraído de [114]. Se parte del conjunto $2^{\mathbb{R}^2}$ formado por todos los subconjuntos dentro del plano $\mathbb{R}^2$ incluido ϕ. Sobre este conjunto se pueden definir las operaciones $\oplus$ como $\cup$ y $\otimes$ como la suma de todos los subconjuntos entre sí:

$$\forall A, B \subseteq \mathbb{R}^2 \ , \ A \otimes B = A + B = \{c \in \mathbb{R}^2 \mid c = a + b, a \in A, b \in B\}$$

Este conjunto es un semianillo idempotente completo dado por la relación de orden $\subseteq$. En la figura B.7, en la imagen de la izquierda, se muestra dos conjuntos: B está representado por el cuadro blanco y A es el circulo gris de forma que $A \subset B$. El resultado de la operación $R_A(B)^{\#} = B/A = \bigvee\{C \in \mathbb{R}^2 | C + A \subseteq B\}$ aparece en la figura central como un cuadro gris, las líneas discontinuas muestran como se ha llegado a ese resultado. En imagen de la derecha se muestra el resultado de la operación $B/A + A$ como un cuadro gris, se puede comprobar como $B/A + A \subseteq B$.

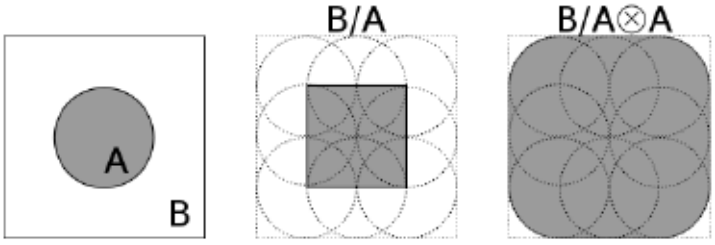


Figura B.7: Ejemplo de la operación / en $\left(2^{\mathbb{R}^2}, \cup, +\right)$

B.4.5 Semianillos y Semimodulos Opuestos

Se puede definir el **opuesto** de un $\mathcal{K}$-*semimódulo* por la izquierda como $\mathcal{M}^{op}$. Este $\mathcal{K}$-*semimódulo* $\mathcal{M}^{op}$ tiene una relación de orden inversa a $\mathcal{M}$ ya que la operación de suma se invierte $m_1 \overset{op}{\oplus} m_2 = m_1 \wedge m_2$. Además la operación de multiplicación será por la derecha

$m \overset{op}{\otimes} k$ y vendrá dada por la *pseudoinversa* de B,4,10, es decir se cumple $m \overset{op}{\otimes} k = k \setminus m$. En cambio para un $\mathcal{K}$-*semimódulo* por la derecha el opuesto será un semimódulo por la izquierda con la operación de multiplicación $k \overset{op}{\otimes} m = m/k$ obtenida de B,4,14.

Si $\mathcal{M}^{op}$ es un $\mathcal{K}$-*semimódulo* por la izquierda se puede definir sus residuos $\overset{op}{\setminus}$ y $\overset{op}{/}$ para el operador $\overset{op}{\otimes}$:

$$\begin{aligned} &\left(L_k^{M^{op}}\right)^{\#} : M \to M \\ &k \overset{op}{\setminus} m_1 = \left(L_k^{M^{op}}\right)^{\#}(m_1) = \bigwedge \{m_2 \in M | m_2/k \geq m_1\} = m_1 \otimes k \end{aligned} \tag{B.4.16}$$

$$\begin{aligned} &\left(R_m^{M^{op}}\right)^{\#} : M \to K \\ &m_1 \overset{op}{/} m_2 = \left(R_{m_2}^{M^{op}}\right)^{\#}(m_1) = \bigvee \{k \in K | m_2/k \geq m_1\} = m_1 \setminus m_2 \end{aligned} \tag{B.4.17}$$

La relación que existe entre los operadores $\overset{op}{\setminus}$, $\overset{op}{/}$ y $\overset{op}{\otimes}$ se puede obtener de forma trivial:

$$\begin{array}{ccccc} k \overset{op}{\otimes} m_2 \geq m_1 & \iff & k \leq m_1 \overset{op}{/} m_2 & \iff & m_2 \geq k \overset{op}{\setminus} m_1 \\ m_2/k \geq m_1 & \iff & k \leq m_1 \setminus m_2 & \iff & m_2 \geq m_1 \otimes k \end{array} \tag{B.4.18}$$

Si suponemos que $\mathcal{M}^{op}$ es un $\mathcal{K}$-semimódulo por la derecha tenemos:

$$\begin{aligned} &\left(L_m^{M^{op}}\right)^{\#} : M \to K \\ &m_2 \overset{op}{\setminus} m_1 = \left(L_{m_2}^{M^{op}}\right)^{\#}(m_1) = \bigvee \{k \in K | k \setminus m_2 \geq m_1\} = m_2/m_1 \end{aligned} \tag{B.4.19}$$

$$\begin{aligned} &\left(R_k^{M^{op}}\right)^{\#} : M \to M \\ &m_1 \overset{op}{/} k = \left(R_k^{M^{op}}\right)^{\#}(m_1) = \bigwedge \{m_2 \in M | k \setminus m_2 \geq m_1\} = k \otimes m_1 \end{aligned} \tag{B.4.20}$$

Al igual que en el caso del semimódulo por la izquierda se puede obtener la relación de los operadores $\overset{op}{\setminus}$, $\overset{op}{/}$ y $\overset{op}{\otimes}$ por la derecha de forma trivial:

$$\begin{array}{ccccc} m_2 \overset{op}{\otimes} k \geq m_1 & \iff & k \leq m_2 \overset{op}{\setminus} m_1 & \iff & m_2 \geq m_1 \overset{op}{/} k \\ k/m_2 \geq m_1 & \iff & k \leq m_2/m_1 & \iff & m_2 \geq k \otimes m_1 \end{array} \tag{B.4.21}$$

Se puede ver fácilmente que tanto $\left(L_k^{M^{op}}\right)^{\#}$ como $\left(R_k^{M^{op}}\right)^{\#}$ son **funciones involutivas** :

$$\left(L_k^{M^{op}}\right)^{\#} \circ \left(L_k^{M^{op}}\right)^{\#}(m_1) = \left(L_k^{M^{op}}\right)^{\#}(m_1 \otimes k) = m_1$$

$$\left(R_k^{M^{op}}\right)^{\#} \circ \left(R_k^{M^{op}}\right)^{\#}(m_1) = \left(R_k^{M^{op}}\right)^{\#}(k \otimes m_1) = m_1$$

Por lo tanto el opuesto de un $\mathcal{K}$-*semimódulo* es una *involución*: $(\mathcal{M}^{op})^{op} = \mathcal{M}$

Un ejemplo de semimódulo opuesto podría ser el $(\mathcal{B},\mathcal{B})$-bisemimódulo booleano $\mathcal{B} = \langle \mathbb{B}, \overset{op}{\oplus}, \overset{op}{\otimes}, 1, 0\rangle$ tal y como se muestra en ([12] ej.5). Es un bisemimódulo completo donde $\bot = \epsilon_{\mathcal{B}^{op}} = 1$ y $\top = e_{\mathcal{B}^{op}} = 0$ la suma se define como:

$$a \overset{op}{\oplus} b = a \wedge b$$

$$0 \wedge 1 = 1 \wedge 0 = 0 \wedge 0 = 0 \qquad 1 \wedge 1 = 1$$

La acción se puede calcular como:

$$a \overset{op}{\otimes} b = a \setminus b = b/a = \bigvee \{c \in B \mid b \otimes c \leq a\} = b \vee a^{op}$$

$$1 \overset{op}{\otimes} 0 = 0 \qquad 0 \overset{op}{\otimes} 0 = 0 \overset{op}{\otimes} 1 = 1 \overset{op}{\otimes} 1 = 1$$

B.4.6 Par dual

Un **par predual** [133] está formado por un $\mathcal{K}$-semimódulo $\mathcal{X}$ por la izquierda completo junto con un $\mathcal{K}$-semimódulo $\mathcal{Y}$ por la derecha completo, juntos bajo la operación $X \times Y$ forman un $\mathcal{K}$-bisemimódulo $\mathcal{Z}$ completo de tal forma que se establecen las siguientes funciones bajo el operador $\langle \cdot \mid \cdot \rangle$:

$$\begin{aligned} R_y : X &\to Z & \qquad L_x : X &\to Z \\ x &\to \langle x \mid y \rangle & \qquad y &\to \langle x \mid y \rangle \end{aligned}$$

Donde R_y es una función continua linear por la izquierda al cumplir:

$$R_y(\lambda \otimes x_1 \oplus \mu \otimes x_2) = \lambda \otimes R_y(x_1) \oplus \mu \otimes R_y(x_2)$$

y de forma dual L_x es una función continua linear por la derecha.

Se dice que $\mathcal{Y}$ *separa a* $\mathcal{X}$ si:

$$(\forall y \in Y, \langle x_1 \mid y \rangle = \langle x_2 \mid y \rangle) \Rightarrow x_1 = x_2$$

y que $\mathcal{X}$ *separa a* $\mathcal{Y}$ si:

$$(\forall x \in X \langle x \mid y_1 \rangle = \langle x \mid y_2 \rangle) \Rightarrow y_1 = y_2$$

Un par predual $(\mathcal{X}, \mathcal{Y})$ tal que X separa Y e Y separa X es un **par dual**.

Si se tiene un $\mathcal{K}$-semimódulo, $\mathcal{X}$, por la derecha, entonces los semimódulos $\mathcal{X}^{op}$ y $\mathcal{X}$ forman un par dual tal que:

$$\begin{aligned} X^{op} \times X &\to K^{op} \\ (y, x) &\mapsto \langle y \mid x \rangle = x \setminus y \end{aligned}$$

Dada la operación $\langle \cdot \mid \cdot \rangle$ de $\mathcal{X} \times \mathcal{Y}$ a un $\mathcal{K}$-semimódulo completo Z y para un pivote $\varphi \in Z$, se definen la funciones:

$$i_l : X \to Y, x \mapsto x^{-} = L_x^{\#}(\varphi) = \bigvee \{y \in Y \mid \langle x \mid y \rangle \leq \varphi\} \tag{B.4.22a}$$

$$i_r : Y \to X, y \mapsto {}^{-}y = R_y^{\#}(\varphi) = \bigvee \{x \in X \mid \langle x \mid y \rangle \leq \varphi\} \tag{B.4.22b}$$

Si (X, Y) es un par predual, entonces:

$$({}^{-}y)^{-} \geq y \qquad {}^{-}\left(({}^{-}y)^{-}\right) = {}^{-}y \qquad \forall y \in Y \tag{B.4.23}$$

$${}^{-}(x^{-}) \geq x \qquad ({}^{-}(x^{-}))^{-} = x^{-} \qquad \forall x \in X \tag{B.4.24}$$

$$y \leq x^{-} \iff \langle x \mid y \rangle \leq \varphi \iff x \leq {}^{-}y \tag{B.4.25}$$

Como se ha visto en la sección B.3.2 las ecuaciones B.4.22 son los polares de una relación contravariante de Galois tal y como queda definido por B.4.25, de forma que $i_l^{\#} = i_r$. Así pues (i_l, i_r) es una adjunción entre X e Y^{op} donde:

$$i_l : \langle X, \leq \rangle \to \langle Y, \overset{op}{\leq} \rangle \qquad\qquad i_r : \langle Y, \leq \rangle) \to \langle X, \overset{op}{\leq} \rangle$$

Los elementos de cierre de X e Y para los operadores ${}^{-}y$ y x^{-} son:

$$\overline{X} = \{{}^{-}y \mid y \in Y\} \qquad\qquad \overline{Y} = \{x^{-} \mid x \in X\} \tag{B.4.26}$$

Como i_l es una función residuada (B.1.6) conserva los máximos (B.3.2). Así que $\overline{Y}$ también es un semi-retículo superior tal y como se definió en B.2.1. Pero el orden de $\langle Y, \overset{op}{\leq} \rangle$ está invertido, así que se convierte en un semi-retículo inferior para el orden normal. De esta forma entre Y e $\overline{Y}$ solo se conservan los mínimos y no los máximos. El mismo razonamiento se puede aplicar a i_r y $\overline{X}$. En particular, la función $x \mapsto x^{-}$ es un *anti-isomorfismo* de retículos completos $\overline{X} \to \overline{Y}$ con inverso $y \mapsto {}^{-}y$.

Un semianillo completo idempotente $\mathcal{K}$ es **reflexivo** si cumple:

$$\forall a \in K: \quad {}^{-}(a^{-}) = a\,, \quad ({}^{-}a)^{-} = a \tag{B.4.27}$$

El elemento φ no tiene que ser único. Si $\mathcal{K}$ es *reflexivo* por la izquierda (o derecha) y k tiene inversa, entonces $\mathcal{K}$ también es reflexivo por la izquierda (o derecha) para $\varphi \otimes k$ (y $k \otimes \varphi$). De esta forma se dirá que $(\mathcal{K}, \varphi)$ es reflexivo.

Por ejemplo, para dos semimódulos $\mathcal{X}_m \simeq \mathcal{K}^{m\times 1}$ y $\mathcal{Y}_m \simeq \mathcal{K}^{1\times m}$ se puede definir [12, ej. 6]:

$$\langle x \mid y \rangle = \bigoplus_i x(i) \otimes y(i)$$

que es un *par predual*. Entonces $\psi_m \overset{def}{=} (i_l, i_r)_m : X_m \multimap Y_m$ es una conexión de Galois para cada valor finito de m.

Para un semianillo completo reflexivo $\mathcal{K}$ y los semimódulos $\mathcal{X} = \mathcal{K}^{1\times n}$ y $\mathcal{Y} = \mathcal{K}^{p\times 1}$, para cada matriz $R \in \mathcal{K}^{n\times p}$ dichos semimódulos forman el par predual $\langle x \mid y \rangle_R = x \otimes R \otimes y$ [133, §4,5]. Partiendo de la ecuación B.4.22 se definen las *(funciones) polares* para un valor específico de $\varphi \in K$ [12, ej. 6], ej.6:

$$x_{\varphi}^{-} = \bigvee \{y \in Y \mid x \otimes R \otimes y \leq \varphi\} = (x \otimes R) \setminus \varphi \tag{B.4.28a}$$

$${}^{-}_{\varphi}y = \bigvee \{x \in X \mid x \otimes R \otimes y \leq \varphi\} = \varphi / (R \otimes y) \tag{B.4.28b}$$

Además, partiendo de la ecuación (B.4.26) se pueden definir los elementos de cierre como:

$$\overline{Y} = \{(x \otimes R) \setminus \varphi \mid x \in \mathcal{K}^{1\times n}\} \tag{B.4.29a}$$

$$\overline{X} = \{\varphi / (R \otimes y) \mid y \in \mathcal{K}^{p\times 1}\} \tag{B.4.29b}$$

de forma que se establece la siguiente conexión de Galois:

$$\psi_m \overset{def}{=} (\cdot^{-}_{\varphi}, {}^{-}_{\varphi}\cdot)_m : \mathcal{K}^{1\times n} \multimap \mathcal{K}^{p\times 1}$$

que es la base del $\mathcal{K}$-Formal Concept Analysis explicado en las Sección 1.4.

C

WebgeneKFCA: Una Herramienta de GED con FCA

C.1 Introducción

Con el fin de crear un marco general para el análisis $\mathcal{K}$-FCA de datos de expresión genética se decidió crear una web que mostrara los principales puntos expuestos en esta tesis. Esta web permite realizar un análisis completo de matrices de expresión de genes, siguiendo la metáfora LofK, donde el usuario puede ir explorando la información contenida en las matrices de expresión. A esta web se pueden subir matrices de expresión genética, aplicarles un preprocesado y analizar con las herramientas visuales que se proporcionan. Además se ha publicado su código en `https://github.com/calabozo/webgenekfca`.

C.2 Inserción de datos

Webgene$\mathcal{K}$-FCA permite tres formas de introducir los datos:

- **Ficheros CEL:** Ficheros de Affymetrix con extensión CEL[1]. Cada fichero contiene el nivel de expresión sin procesar de diferentes sondas. Una vez en el servidor se procesarán todos los ficheros juntos y generarán una matriz de expresión con *apt-probeset-summarize*[2].
- **Fichero texto**: Fichero con valores separados por comas, espacios en blanco o tabuladores. Cada columna muestra el nivel de expresión de un experimento, cada fila el nivel de expresión de una sonda o gen. Se puede usar como fichero de entrada la salida obtenida de *apt-probeset-summarize*.
- **Ficheros de conteo**: Se utiliza para análisis RNA-Seq, consiste en ficheros con dos columnas, la primera contiene el nombre del gen, la segunda el número de veces que se ha podido contar una ocurrencia de dicho gen en la muestra. La idea es utilizar

[1] `http://media.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/cel.html`
[2] `http://media.affymetrix.com/support/developer/powertools/changelog/apt-probeset-summarize.html`

htseq-count[3] para generar este tipo de ficheros. Habrá tantos ficheros como muestras se deseen analizar.

Además de los ficheros de datos se puede añadir más información, como un nombre para definir el análisis y una breve descripción. Así mismo es necesario indicar qué tipo de organismo se está analizando, lo que se utilizará para conectar con diferentes bases de datos y enriquecer los resultados del análisis con información extra que ayudará al investigador a aumentar su conocimiento de los genes (cfr. Sección 2.2).

C.3 Preprocesado

Una vez que se ha subido una matriz de expresión es necesario realizar un preprocesado como el indicado en la Sección 3.2. Una captura de pantalla del preprocesado se puede ver en la figura C.1.

En la primera imagen aparece el histograma del nivel de expresión genética que cambiará en función de la ecuación de normalización seleccionada con botones de radio directamente bajo ella (cfr. § 3.2.2 para los diferentes tipos de normalización).

Si se está analizando un microarray de Affymetrix también se podrá seleccionar si se quieren mostrar directamente el valor de la sondas (probesets) o combinarlos para mostrar el nivel de expresión genética como se indica en 3.2.1.1. Si el análisis se basa en una matriz obtenida mediante RNA-Seq esto no aparece porque se tendrá directamente el conteo de los genes.

C.4 Inicio del estudio

Una vez acabada la parte del preprocesado se puede empezar con el estudio de los datos para tratar de extraer la máxima información posible. Por esto se han creado dos vistas diferentes para realizar: *1*) la exploración de datos con $\mathcal{K}$-FCA y *2*) su análisis. Gracias a estas herramientas visuales el usuario podrá ver los datos desde diferentes perspectivas lo cual le permitirá identificar

C.4.1 Exploración $\mathcal{K} - FCA$

En esta página se puede ver el número de conceptos en función de φ/ϕ tal y como se ha mostrado por ejemplo en la figura 5.1. Además se pueden realizar las siguientes acciones:

- Exploración $\mathcal{K}$-FCA: Se puede seleccionar el umbral de φ/ϕ que se desea tener para el dominio en maxplus o minplus, respectivamente. Cada vez que se selecciona un umbral en la barra de deslizamiento de la izquierda aparece un nuevo retículo.
- Seleccionar concepto formal del retículo: Esto permite estudiar los genes, identificar el filtro e ideal, calcular el "p-value" basándose en términos GO.

[3] http://www-huber.embl.de/users/anders/HTSeq/doc/count.html

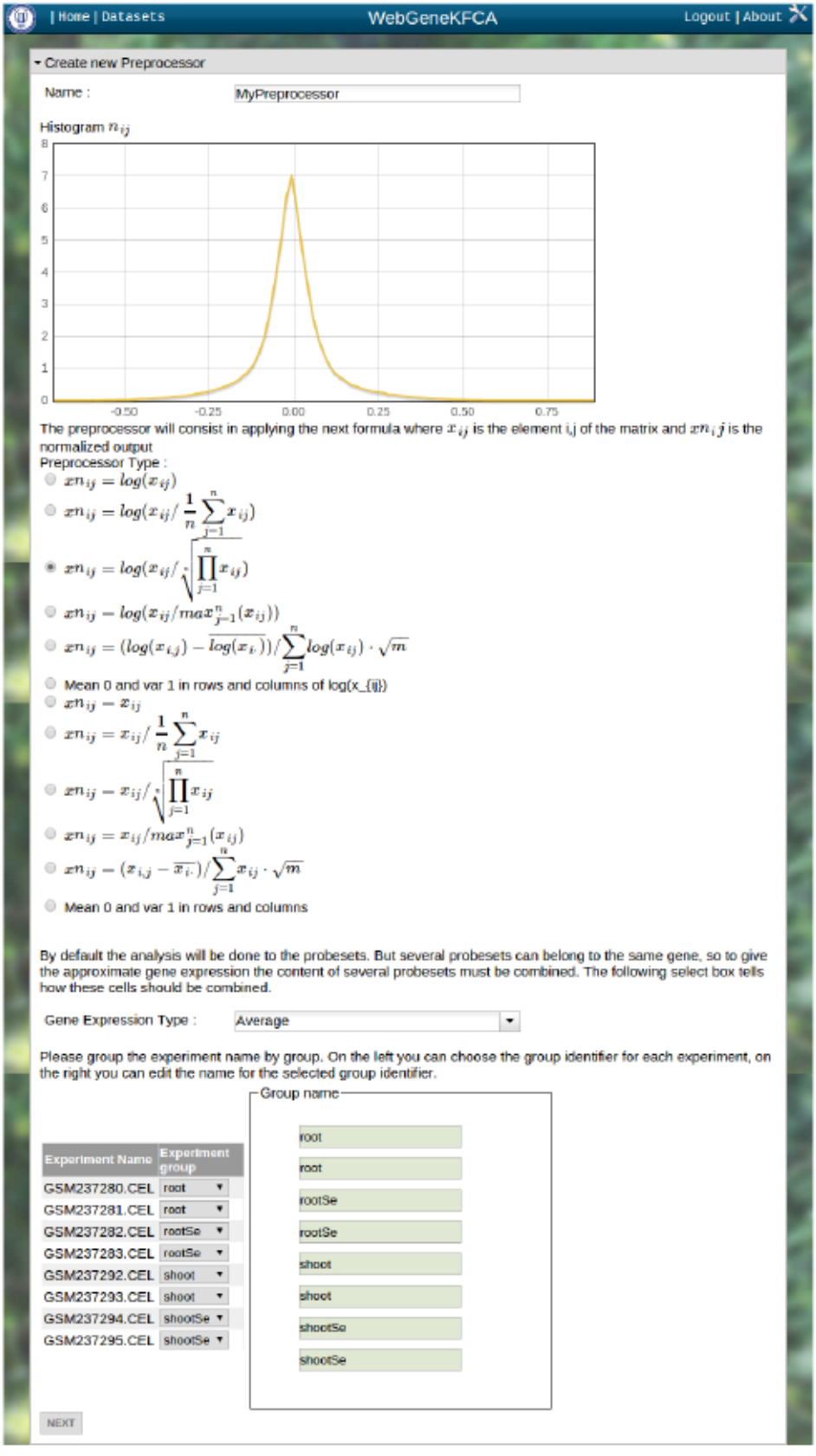


Figura C.1: Pantalla de preprocesado de *webgeneKFCA*.

- Seleccionar gen: Al seleccionar un gen se pueden ver sus diferentes nombres, una breve descripción de su funcionalidad, enlaces a otras webs donde se puede ampliar la información acerca del gen y además se puede ver la evolución del retículo estructural en función del umbral.

C.4.2 Análisis $\mathcal{K} - FCA$

Desde esta otra vista, un ejemplo de la cual se muestra en la figura C.2, se pueden buscar los genes que pertenecen exclusivamente a un concepto y ver cómo dichos genes van entrando y saliendo del grupo en función del umbral de una forma similar a como se explica en la Sección 3.4.3.

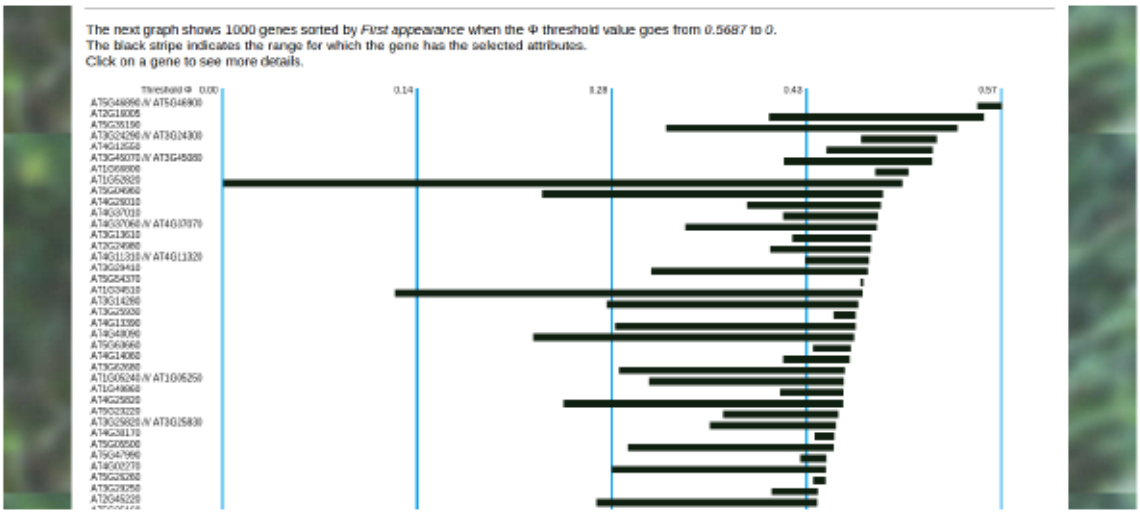


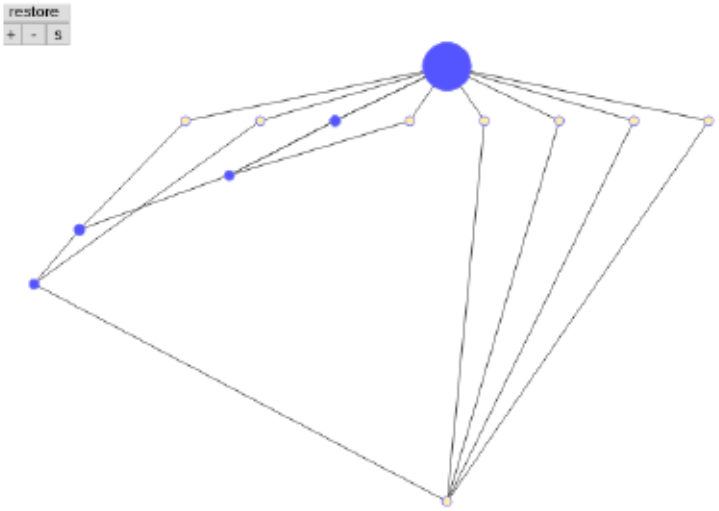
Figura C.2: Pantalla donde se muestra el rango de pertenencia de un gen a un grupo.

En ella es posible seleccionar los atributos que se quieren tener activos, el álgebra a utilizar (maxplus o minplus) y el orden en el cual se desea que aparezcan los genes: por primera aparición o por un mayor rango de existencia en el grupo. Al seleccionar todos los elementos de búsqueda aparece una vista que permite ver el paso de los genes a través de un grupo seleccionado.

C.4.3 Información extra

C.4.3.1 Descripción detallada de un gen

Además de las vistas anteriores existe una tercera, mostrada en la figura C.3 y a la cual se puede llegar desde las vistas ya descritas, en donde se da más información acerca del gen o sonda seleccionada. En esta vista se puede ver toda la información técnica disponible para dicho gen como su título, identificador en diferentes bases de datos (AGI, FlyBase), referencias a GO y a Kyoto Encyclopedia of Genes and Genomes (KEGG), junto con enlaces que llevan a webs donde se puede ampliar la información de dicho gen.



Probe Set ID	262517_at
Gene Title	ATGSTU25 (GLUTATHIONE S-TRANSFERASE TAU 25); glutathione transferase
Target Description	putative glutathione transferase Second of three repeated putative glutathione transferases. 72% identical to glutathione transferase [Arabidopsis thaliana] (gi|4006934). Location of ests 191A10T7 (gb|R90188) and 171N13T7 (gb|R65532)
AGI	AT1G17180
Gene Symbol	ATGSTU25
QTL	
Gene Ontology Biological Process	Accession: GO:0009407 — Definition: toxin catabolic process
Gene Ontology Cellular Component	Accession: GO:0005737 — Definition: cytoplasm
Gene Ontology Molecular Function	Accession: GO:0004364 — Definition: glutathione transferase activity Accession: GO:0016740 — Definition: transferase activity
Pathway	---
GeneChip Array	
Transcript ID(Array Design)	At1g17180
Species Scientific Name	
Annotation Date	Jun 9, 2011
Sequence Type	Exemplar sequence
Sequence Source	Affymetrix Proprietary Database
Representative Public ID	At1g17180
Archival UniGene Cluster	---
UniGene ID	At.41849
Genome Version	June 2009 (TAIR 9)
Alignments	Chr1:5872207-5872958 (+) // 100.0 //
Gene Symbol	ATGSTU25

Figura C.3: Vista detallada de la descripción de un gen.

Además de la información obtenida de base de datos, se ofrece una vista de un retículo con todos los atributos que tiene el gen analizado en el rango de la exploración. Esto permite ver cómo evoluciona un gen según los umbrales, como se muestra en la figura 5.4 y se explica en la sección 3.4.3. Un ejemplo de toda la información que proporciona esta vista se encuentra en la figura 5.11. Hay información que no aparece en la imagen, como el rango de valores de ϕ para el cual el gen está en ese concepto que sólo aparece cuando el puntero se mueve por encima de un concepto.

C.4.3.2 Ontología genética

Es posible estimar la calidad de un grupo mirando los genes seleccionados y sus términos GO tal y como se explica en la sección 3.6. Además desde aquí se puede uno bajar un fichero en formato CSV para ser usado en Conexp que permite ver la relación que guardan entre si todos los términos GO como se muestra en la figura 5.3.

C.5 Datos técnicos

La web dispone de un sistema de identificación que permite tener usuarios con rol de administrador. Estos usuario son los únicos que pueden realizar operaciones de escritura en la base de datos, lo que significa que pueden subir experimentos y generar nuevos resultados. También disponen de acceso a un sistema de autodiagnóstico que permite ver la carga de CPU, el uso de memoria, el uso de disco e información sobre las bases de datos importadas. Un usuario invitado puede ver todos los experimentos subidos y obtener todas las gráficas que se han enseñado en esta tesis.

Como se ha dicho anteriormente esta tesis se basa no sólo en el análisis de datos utilizando $\mathcal{K}$-FCA sino que para llevarlo a cabo se apoya en diferentes bases de datos externas que ayuda a confeccionar un mapa de conocimiento de mejor calidad. Los sistemas externos con los que se conecta el servidor son los siguientes:

Conexión con sistemas externos:

- Gene Ontology: `http://geneontology.org`
- Kyoto Encyclopedia of Genes and Genomes: `http://www.genome.jp/kegg`
- National Center for Biotechnology Information: `http://www.ncbi.nlm.nih.gov`
- Arabidopsis Genome Initiative: `http://www.arabidopsis.org`
- FlyBase: `http://flybase.org`

Para hacer funcionar el sistema hace falta PC con las siguientes características:

Requisitos técnicos:

- Java 1.7
- Tomcat 7

- Mysql 5
- Apt-affymetrix tools
- Al menos 1GB de disco libre
- Al menos 2GB de RAM

Bibliografía

[1] J. W. Tukey, "We need both exploratory and confirmatory," *Am. Stat.*, vol. 34, no. 1, pp. 23–25, 1980.

[2] F. Crick, "Central dogma of molecular biology.," 1970.

[3] S. Zhao, W.-P. Fung-Leung, A. Bittner, K. Ngo, and X. Liu, "Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells.," *PLoS One*, vol. 9, no. 1, p. e78644, 2014.

[4] G. B. Stefano, "Comparing Bioinformatic Gene Expression Profiling Methods: Microarray and RNA-Seq," *Med. Sci. Monit. Basic Res.*, vol. 20, pp. 138–142, 2014.

[5] D. R. Brillinger, "Data Analysis, Exploratory," *Int. Encycl. Polit. Sci.*, no. 2006, pp. 530–537, 2011.

[6] B. Mirkin, "Mathematical classification and clustering," 1996.

[7] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Berlin, Heidelberg: Springer, 1999.

[8] U. Priss, "Formal Concept Analysis in Information Science," *Rev. Inf. Sci.*, vol. 40, no. 1989, pp. 1–22, 2002.

[9] B. Ganter and R. Wille, "Applied lattice theory: formal concept analysis," *Prepr. http//wwwbib. Math. tudarmstadt. de/Math-Net/Preprints/Listen/pp97. html*, 1997.

[10] S. Yevtushenko, "System of data analysis Çoncept Explorer". (In Russian)," *Proc. 7th Natl. Conf. Artif. Intell.*, pp. 127–134, 2000.

[11] Priestley and Davey, "Lattices and Complete Lattices," in *Introd. to Lattices Order*, ch. 2, p. 310, Cambridge University Press, 2002.

[12] F. J. Valverde-albacete, "Towards a Generalisation of Formal Concept Analysis for Data Mining purposes," *Analysis*, vol. 3874, 2006.

[13] F. J. Valverde-Albacete and C. Peláez-Moreno, "Galois Connections between Semimodules and Applications in Data Mining," *Form. Concept Anal.*, pp. 181–196, 2007.

[14] R. Wille, "Conceptual Landscapes of Knowledge: A Pragmatic Paradigm for Knowledge Processing," in *Classif. Inf. Age*, pp. 344–356, Dresden: Springer, 1998.

[15] G. Lakoff and M. Johnsen, *Metaphors we live by*. The university of Chicago press, 1996.

[16] G. Lakoff, *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, vol. 64. Linguistic Society of America, 1987.

[17] F. J. Valverde-Albacete, J. M. Gonzalez-Calabozo, A. Peñas, and C. Pelaez-Moreno, "Supporting Scientific Knowledge Discovery with Extended, Generalized Formal Concept Analysis," *Expert Syst. Appl.*, 2015.

[18] C. Peláez-Moreno, A. I. García-Moral, and F. J. Valverde-Albacete, "Analyzing phonetic confusions using Formal Concept Analysis," *Journal of the Acoustical Society of America*, vol. 128, pp. 1377–1390, Sep 2010.

[19] C. Pelaez-Moreno, A. I. Garcia-Moral, and F. J. Valverde-Albacete, "Eliciting a hierarchical structure of human consonant perception task errors using formal concept analysis," *Interspeech 2009: 10th Annual Conference of the International Speech Communication Association*, vol. 1–5, pp. 824–827, 2009. 10th INTERSPEECH 2009 Conference SEP 06-10, 2009 Brighton, ENGLAND.

[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. fifth Berkeley Symp. . . .*, vol. 233, no. 233, pp. 281–297, 1967.

[21] I. S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," *Proc. 7 th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. pp, pp. 269–274, 2001.

[22] J. A. Hartigan, "Direct Clustering of Data Matrix," *J. Am. Stat. Assoc.*, vol. 67, 1972.

[23] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. van Sanden, D. Lin, W. Talloen, L. Bijnens, H. W. H. Göhlmann, Z. Shkedy, and D. A. Clevert, "FABIA: Factor analysis for bicluster acquisition," *Bioinformatics*, vol. 26, no. 12, pp. 1520–1527, 2010.

[24] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey.," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 1, no. 1, pp. 24–45, 2004.

[25] Y. Cheng and G. M. Church, "Biclustering of Expression Data," in *Proc. Eighth Int. Conf. Intell. Syst. Mol. Biol.*, pp. 93–103, 2000.

[26] M. C. P. de Souto, I. G. Costa, D. S. a. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: a comparative study.," *BMC Bioinformatics*, vol. 9, p. 497, jan 2008.

[27] N. Gupta and S. Aggarwal, "MIB: Using mutual information for biclustering gene expression data," *Pattern Recognit.*, 2010.

[28] G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 97, no. 22, pp. 12079–12084, 2000.

[29] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter, "Furby: fuzzy force-directed bicluster visualization.," *BMC Bioinformatics*, vol. 15 Suppl 6, p. S4, jan 2014.

[30] C. Tang and L. Zhang, "Interrelated two-way clustering: an unsupervised approach for gene expression data analysis," *. . . , 2001. Proc. . . .*, pp. 41–48, 2001.

[31] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data.," *Bioinformatics*, vol. 22, no. 9, pp. 1122–9, 2006.

[32] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: coclustering genes and conditions.," *Genome Res.*, vol. 13, no. 4, pp. 703–16, 2003.

[33] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem.," *J. Comput. Biol.*, vol. 10, pp. 373–84, jan 2003.

[34] L. Lazzeroni and A. Owen, "Plaid Models for Gene Expression Data," *Stat. Sin.*, vol. 12, pp. 61–86, 2002.

[35] H. Turner, T. Bailey, and W. Krzanowski, "Improved biclustering of microarray data demonstrated through systematic performance tests," *Comput. Stat. Data Anal.*, vol. 48, pp. 235–254, feb 2005.

[36] A. Tanay, R. Sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data.," *Bioinformatics*, vol. 18 Suppl 1, pp. S136–S144, 2002.

[37] a. Abdullah and A. Hussain, "A new biclustering technique based on crossing minimization," *Neurocomputing*, vol. 69, no. 16-18, pp. 1882–1896, 2006.

[38] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," *Proceeding Elev. ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '05*, p. 41, 2005.

[39] F. Domenach and B. Leclerc, "The structure of the overhanging relations associated with some types of closure systems," *Annals of Mathematics and Artificial Intelligence*, vol. 49, no. 1–4, pp. 137–149, 2007.

[40] D. B. R. A. P. Dempster, N. M. Laird, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[41] I. Van Mechelen, H.-H. Bock, and P. De Boeck, "Two-mode clustering methods: a structured overview.," *Stat. Methods Med. Res.*, vol. 13, pp. 363–94, oct 2004.

[42] J. Liu, Z. Li, X. Hu, and Y. Chen, "Biclustering of microarray data with MOSPO based on crowding distance.," *BMC Bioinformatics*, vol. 10 Suppl 4, p. S9, 2009.

[43] E. Masciari, G. Mazzeo, and C. Zaniolo, "Analysing microarray expression data through effective clustering," *Inf. Sci. (Ny).*, dec 2013.

[44] Y. Si, P. Liu, P. Li, and T. P. Brutnell, "Model-based clustering for RNA-seq data.," *Bioinformatics*, vol. 30, pp. 197–205, jan 2014.

[45] R. Irizarry, C. Wang, Y. Zhou, and T. P. Speed, "Gene set enrichment analysis made simple," *Stat. Methods Med. Res.*, vol. 18, no. 6, pp. 565–575, 2009.

[46] L. Scheubert, *Selecting Biomarkers for Pluripotency and Alzheimers Disease*. PhD thesis, Fachbereich Mathematik/Informatik - Universität Osnabrück, 2012.

[47] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture.," *Nat. Genet.*, vol. 22, pp. 281–5, jul 1999.

[48] G. F. Berriz, O. D. King, B. Bryant, C. Sander, and F. P. Roth, "Characterizing gene sets with FuncAssociate," *Bioinformatics*, vol. 19, no. 18, pp. 2502–2504, 2003.

[49] M. Ashburner, C. a. Ball, J. a. Blake, D. Botstein, H. Butler, J. M. Cherry, a. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. a. Harris, D. P. Hill, L. Issel-Tarver, a. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.," *Nat. Genet.*, vol. 25, pp. 25–9, may 2000.

[50] C. Carpineto and G. Romano, *Concept Data Analysis: Theory and Applications*. Wiley, 2004.

[51] R. Pensa, J. Besson, and J.-F. Boulicaut, "A methodology for biologically relevant pattern discovery from gene expression data," in *Discovery Science* (E. Suzuki and S. Arikawa, eds.), vol. 3245 of *Lecture Notes in Computer Science*, pp. 230–241, Springer Berlin Heidelberg, 2004.

[52] M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis, "Mining gene expression data with pattern structures in formal concept analysis," *Inf. Sci. (Ny).*, no. August, 2010.

[53] M. Kaytoue, S. Duplessis, S. O. Kuznetsov, and A. Napoli, *Two FCA-Based Methods for Mining Gene*. Springer Berlin Heidelberg, 2009.

[54] S. Motameny, B. Versmold, and R. Schmutzler, "Formal Concept Analysis for the Identification of Combinatorial Biomarkers in Breast Cancer," *Gene*, pp. 229–240, 2008.

[55] J. Gebert, S. Motameny, U. Faigle, C. V. Forst, and R. Schrader, "Identifying Genes of Gene Regulatory Networks Using Formal Concept Analysis," *J. Comput. Biol.*, vol. 15, no. 2, pp. 185–194, 2008.

[56] S. Kuznetsov, "Galois connections in data analysis: Contributions from the soviet era and modern russian research," *Formal Concept Analysis*, 2005.

[57] S. Kuznetsov, "Pattern structures for analyzing complex data," *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, vol. LNCS, 5908, pp. 33–44, 2009.

[58] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene, "Fuzzy and rough formal concept analysis: a survey.," *Int. J. General Systems ()*, vol. 43, no. 2, pp. 105–134, 2014.

[59] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene, "Formal concept analysis in knowledge processing: A survey on applications," *Expert Systems with Applications*, vol. 40, pp. 6538–6560, Nov. 2013.

[60] J. Poelmans, S. O. Kuznetsov, D. I. Ignatov, and G. Dedene, "Formal Concept Analysis in Knowledge Processing: A Survey on Models and Techniques," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6601–6623, 2013.

[61] J. Poelmans, D. Ignatov, S. Viaene, G. Dedene, and S. Kuznetsov, "Text mining scientific papers: a survey on FCA-based information retrieval research," in *Advances in Data Mining Applications and Theoretical Aspects, ICDM 2012* (P. Perner, ed.), pp. 273–287, Springer, 2012.

[62] F. J. Valverde-Albacete and C. Peláez-Moreno, "Extending conceptualisation modes for generalised Formal Concept Analysis," *Information Sciences*, vol. 181, pp. 1888–1909, May 2011.

[63] R. Bělohlávek, *Fuzzy Relational Systems. Foundations and Principles*, vol. 20 of *IFSR International Series on Systems Science and Engineering*. Kluwer Academic, 2002.

[64] J. M. Gonzalez-Calabozo, C. Peláez-Moreno, and F. J. Valverde-Albacete, "WebGeneKFCA : an On-line Conceptual Analysis Tool for Genomic Expression Data," *Concept Lattices Appl.*, pp. 345–350, 2012.

[65] J. M. Gonzalez-Calabozo, C. Pelaez-Moreno, and F. J. Valverde-albacete, "Gene Expression Array Exploration Using K -Formal Concept Analysis," in *ICFCA*, vol. 0223, (Nicosia, Cyprus), pp. 119–134, Springer, 2011.

[66] R. a. Irizarry, "Summaries of Affymetrix GeneChip probe level data," *Nucleic Acids Res.*, vol. 31, pp. 15e–15, feb 2003.

[67] Z. Liu and X. Zhang, "Effects of Multiple Probesets in Affymetrix GeneChips on Identifying Differentially Expressed Genes in iPS Cells," *Fourth Int. Conf. Comput. Syst. Biol.*, pp. 187–195, 2010.

[68] V. Bolón-Canedo, N. Sánchez-Maroño, a. Alonso-Betanzos, J. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Inf. Sci. (Ny).*, vol. 282, pp. 111–135, oct 2014.

[69] W. Wong, M. Gail, K. Krickeberg, A. Tsiatis, and J. Samet, *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, 2005.

[70] C. Winter, S. Kersting, and J. Roy, "Google Goes Cancer: Improving Outcome Prediction for Cancer Patients by Network-Based Ranking of Marker Genes," *PLoS Comput. Biol.*, vol. 8, no. 5, 2012.

[71] M. a. Busby, J. M. Gray, A. M. Costa, C. Stewart, M. P. Stromberg, D. Barnett, J. H. Chuang, M. Springer, and G. T. Marth, "Expression divergence measured by transcriptome sequencing of four yeast species.," *BMC Genomics*, vol. 12, p. 635, jan 2011.

[72] J. Herrero and A. Valencia, "Network for clustering gene expression patterns," *Bioinformatics*, vol. 17, no. 2, pp. 126–136, 2001.

[73] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey," *IEEE Trans. Knowl. Data Eng.*, vol. 16, pp. 1370–1386, nov 2004.

[74] R. A. Olshen and B. Rajaratnam, "Successive normalization of rectangular arrays," *Ann Stat.*, vol. 38, no. 3, pp. 1638–1664, 2010.

[75] S. Gaubert, "Two lectures on max-plus algebra," *Support cours Ia*, vol. 26, no. March, 1998.

[76] M. Akian, R. Bapat, and S. Gaubert, "Max-Plus Algebra," *Cycle*, vol. 25, pp. 1–18, 2006.

[77] F. J. Valverde-Albacete and C. Peláez-Moreno, "Further Galois Connections between Semimodules over Idempotent Semirings," in *Proc. 4th Conf. Concept Lattices Appl. (CLA 2007)*, Citeseer, 2007.

[78] F. J. Valverde-Albacete and C. Peláez-Moreno, "The spectra of irreducible matrices over completed idempotent semifields," *Fuzzy Sets Syst.*, vol. 271, pp. 46–69, 2015.

[79] P. Eklund and J. Villerd, "A Survey of Hybrid Representation of Concept Lattices in Conceptual Knowledge Processing," *I8th Int. Conf. Form. Concept Anal. - ICFCA 2010*, vol. 5986, pp. 296–311, 2010.

[80] M. Gardner, *Mathematical Carnival*. Mathematical Association of America, 1989.

[81] P. Hansen and B. Jaumard, "Cluster analysis and mathematical programming," *Math. Program.*, vol. 79, pp. 191–215, oct 1997.

[82] I. Gat-Viks, R. Sharan, and R. Shamir, "Scoring clustering solutions by their biological relevance," *Bioinformatics*, vol. 19, pp. 2381–2389, dec 2003.

[83] I. Rivals, L. Personnaz, L. Taing, and M.-C. Potier, "Enrichment or depletion of a GO category within a class of genes: which test?," *Bioinformatics*, vol. 23, pp. 401–7, feb 2007.

[84] R. Godin, C. Pichet, and J. Gecsei, "Design of a browsing interface for information retrieval," *ACM SIGIR Forum*, vol. 23, pp. 32–39, 1989.

[85] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "Data, information, knowledge and principle: back to metabolism in KEGG," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D199–D205, 2014.

[86] X. Cui and G. A. Churchill, "Statistical tests for differential expression in cDNA microarray experiments.," *Genome Biol.*, vol. 4, p. 210, jan 2003.

[87] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Dept. Informatics, Aristotle Univ. ...*, 2006.

[88] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognit.*, vol. 45, no. 9, pp. 3084–3104, 2012.

[89] J. D. Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2012.

[90] D. Dembélé, "A Flexible Microarray Data Simulation Model," *Microarrays*, vol. 2, pp. 115–130, apr 2013.

[91] M. Nykter, T. Aho, M. Ahdesmäki, P. Ruusuvuori, A. Lehmussola, and O. Yli-Harja, "Simulation of microarray data with realistic characteristics.," *BMC Bioinformatics*, vol. 7, p. 349, jan 2006.

[92] L. Ingrid and T. Speed, "Replicated microarray data," *Stat. Sin.*, vol. 12, pp. 31–46, 2002.

[93] M. L. Metzker, "Sequencing technologies - the next generation.," *Nat. Rev. Genet.*, vol. 11, pp. 31–46, jan 2010.

[94] J. Marioni, C. Mason, and S. Mane, "RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays," *Genome . . .*, pp. 1509–1517, 2008.

[95] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of RNA-seq data.," *Genome Biol.*, vol. 11, p. R25, jan 2010.

[96] M. D. Robinson and G. K. Smyth, "Small-sample estimation of negative binomial dispersion, with applications to SAGE data.," *Biostatistics*, vol. 9, pp. 321–32, apr 2008.

[97] J. Gupta, S. Singh, and N. Verma, "MTBA: MATLAB Toolbox for Biclustering Analysis," *MTBA MATLAB Toolbox Biclustering Anal.*, vol. 1, pp. 94–97, 2013.

[98] D. Van Hoewyk, H. Takahashi, E. Inoue, A. Hess, M. Tamaoki, and E. a. H. Pilon-Smits, "Transcriptome analyses give insights into selenium-stress responses and selenium tolerance mechanisms in Arabidopsis.," *Physiol. Plant.*, vol. 132, pp. 236–53, feb 2008.

[99] J. Jiang, Y. Jing, G. J. Cost, J.-C. Chiang, H. J. Kolpa, A. M. Cotton, D. M. Carone, B. R. Carone, D. a. Shivak, D. Y. Guschin, J. R. Pearl, E. J. Rebar, M. Byron, P. D. Gregory, C. J. Brown, F. D. Urnov, L. L. Hall, and J. B. Lawrence, "Translating dosage compensation to trisomy 21," *Nature*, jul 2013.

[100] J. M. Gonzalez-Calabozo, F. J. Valverde-Albacete, and C. Peláez-Moreno, "Interactive knowledge discovery and data mining on genomic expression data with numeric formal concept analysis," *BMC Bioinformatics*, Submitted.

[101] T. Robinson, *Genetics For Dummies*. –For dummies, Wiley, 2005.

[102] J. Pevsner, *Bioinformatics and Functional Genomics*. Wiley, 2009.

[103] R. Dawkins, *The Selfish Gene*. New York: Oxford University Press, 1976.

[104] Z. Su, H. Fang, H. Hong, L. Shi, W. Zhang, W. Zhang, Y. Zhang, Z. Dong, L. J. Lancashire, M. Bessarabova, X. Yang, B. Ning, B. Gong, J. Meehan, J. Xu, W. Ge, R. Perkins, M. Fischer, and W. Tong, "An investigation of biomarkers derived from legacy microarray data for their utility in the RNA-seq era.," *Genome Biol.*, vol. 15, no. 12, p. 523, 2014.

[105] I. Affymetrix, "Statistical algorithms description document," *Affymetrix, St. Cl.*, 2002.

[106] F. Sanger and a. R. Coulson, "A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase.," *J. Mol. Biol.*, vol. 94, pp. 441–8, may 1975.

[107] Illumina, "An Introduction to Next-Generation Sequencing Technology Welcome to Next-Generation Sequencing," tech. rep., illumina, 2014.

[108] M. Ronaghi, "Pyrosequencing Sheds Light on DNA Sequencing," *Genome Res.*, vol. 11, pp. 3–11, jan 2001.

[109] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics.," *Nat. Rev. Genet.*, vol. 10, pp. 57–63, jan 2009.

[110] P. J. a. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants.," *Nucleic Acids Res.*, vol. 38, pp. 1767–71, apr 2010.

[111] T. Sam, B. A. M. Format, and S. Working, "Sequence Alignment / Map Format Specification," *Bioinformatics*, pp. 1–17, 2014.

[112] S. Anders, D. J. McCarthy, Y. Chen, M. Okoniewski, G. K. Smyth, W. Huber, and M. D. Robinson, "Count-based differential expression analysis of RNA sequencing data using R and Bioconductor.," *Nat. Protoc.*, vol. 8, pp. 1765–86, sep 2013.

[113] B. Davey and H. Priestley, *Introduction to lattices and order*. Cambridge, UK: Cambridge University Press, 2nd ed., 2002.

[114] F. Baccelli, G. Cohen, G. J. Olsder, and J.-p. Quadrat, *Synchronization and Linearity*. Wiley, 1992.

[115] A. Blass, S. Burris, and H. P. Sankappanavar, "A Course in Universal Algebra.," *Am. Math. Mon.*, vol. 91, p. 64, jan 1984.

[116] S. Roman, *Lattices and Ordered Sets*. Springer, 2008.

[117] M. Erné, J. Koslowski, A. Melton, and G. Strecker, "A primer on Galois connections," *Ann. N. Y. Acad. Sci.*, vol. 704, no. Papers on General Topology and Applications, pp. 103–125, 2006.

[118] G. Birkhoff, *Lattice Theory*. American Mathematical Society, 1st ed. ed., 1940.

[119] J. Koppitz, "Closure Operators and Lattices," in *M-solid Var. Algebr.*, no. ii in Advances in Mathematics, Springer, 2006.

[120] K. D. M. E. S. Wismath, *Galois Connections and Applications*. Springer, 2004.

[121] O. Ore, *Theory of Graphs*, vol. XXXVIII of *American Mathematical Society Colloquiun Publications*. Providence, Rhode Island: American Mathematical Society, 1962.

[122] G. Birkhoff, *Lattice theory*. American Mathematical Society, 3rd ed. ed., 1967.

[123] M. Barbut and B. Monjardet, *Ordre et classification. Algèbre et combinatoire, tome I*. Méthodes mathématiques des sciences de l'Homme, Hachette, 1970.

[124] M. Barbut and B. Monjardet, *Ordre et classification. Algèbre et combinatoire, tome II*. Méthodes mathématiques des sciences de l'Homme, Hachette, 1970.

[125] L. C. Freeman and D. R. White, "Using Galois Lattices to Represent Nework Data," *Analysis*, vol. 23, no. 1993, pp. 127–146, 2009.

[126] J. S. Golan, *Semirings and Affine Equations over Them: Theory and Applications (Mathematics and Its Applications)*. Kluwer Academic Publishers, 2003.

[127] J. S. Golan, *Semirings and their Applications*. Kluwer Academic, 1999.

[128] J. S. Golan, *Power Algebras over Semirings. With Applications in Mathematics and Computer Science*, vol. 488 of *Mathematics and its applications*. Dordrecht, Boston, London: Kluwer Academic, 1999.

[129] I. Herstein, *Topics in algebra*. Wiley, 1975.

[130] J. Berstel and D. Perrin, *Theory of codes*. Academic Press, 1985.

[131] T. Brunsch, L. Hardouin, C. A. Maia, and J. Raisch, "Duality and interval analysis over idempotent semirings," *Linear Algebra Appl.*, vol. 437, pp. 2436–2454, nov 2012.

[132] R. Howard, "What We Need to Know about Rings and Modules," *Class Notes for Mathematics 700*, pp. 1–10, 1995.

[133] G. Cohen, S. Gaubert, and J.-P. Quadrat, "Duality and separation theorems in idempotent semimodules*1," *Linear Algebra Appl.*, vol. 379, pp. 395–422, dec 2004.

Lista de acrónimos

$\mathcal{K}$-**FCA**	$\mathcal{K}$-Formal Concept Analysis
ADN	Ácido desoxirribonucleico
ADNc	ADN complementario
AGI	Arabidopsis Genome Initiative
ARN	Ácido ribonucleico
ARNm	ARN mensajero
BiMax	Binary inclusion-Maximal biclustering algorithm
CDA	Confirmatory Data Analysis
CDF	Chip Description File
CMOPSOB	Crowding distance based Multi-objective Particle Swarm Optimization Biclustering
cpo	conjunto parcialmente ordenado
CTWC	Coupled Two-Way Clustering
EDA	Exploratory Data Analysis
EM	Expectation Maximization
FABIA	Factor analysis for bicluster acquisition
FCA	Formal Concept Analysis
GED	Gene Expression Data
GO	Gene Ontology
KEGG	Kyoto Encyclopedia of Genes and Genomes
LofK	Landscapes of Knowledge
M-CLUBS	Microarray data CLustering Using Binary Splitting
MIPS	Martinsried Institute of Protein Sciences
NCBI	National Center for Biotechnology Information
NGS	Next-Generation Sequencing
OPSMs	order-preserving submatrices
RMA	robust multi-array analysis

RNA-Seq	RNA sequencing
SOM	Self-organizing map
SPC	superparamagnetic clustering algorithm
SRA	Sequence Read Archive

Índice alfabético